

Oracle Multimaster Replication Maintenance Optimization

Msc. Hakik Paci (PhD Candidate)

Faculty of Information Technology, Polytechnic University of Tirana, Albania

Prof.Dr. Aleksander Xhuvani

Faculty of Information Technology, Polytechnic University of Tirana, Albania

Abstract

This paper presents a new solution for adding a new database server on an existing Oracle Multimaster data replication system. In normal conditions adding a new database server on this system we cannot insert, delete and update data during this time. So during this time the system is down and we can make only queries. The time for adding the new database server depends on the number of tables on the replication group and on the network conditions. To reduce this time we propose to replace the Replication group in more than one replication group based on logic of database schema. So we will have a replication group for base data, one replication group for most populated tables and another group for non most used tables. Using this method on the Oracle database schema with 250 replication tables and communication speed 1Mbit/s, the time reduced seven times. This research is implemented on border control system (Total Information Management System at Ministry of Interior) in Albania and using this solution we can add or remove replication sites with minimal down time.

Key words: Keywords: Oracle Multimaster Data Replication, Replication, Replication group

1. Introduction

Replication is the process of copying and maintaining database objects, such as tables, in multiple databases that make up a distributed database system. Changes applied at one site are stored locally before being forwarded and applied at each of the remote locations.

Some of the most common reasons for using replication are described as follows:

- Availability and Performance

Replication provides fast access to shared data because it enables user access to a local database instead of a remote database. Some users can access one database server while other users can access different database servers which all provide the same data, thereby reducing the workload of all servers.

- Network Load Reduction

Replication can be used to distribute data over multiple regional locations. At that point, applications can access various regional servers instead of accessing one central server. This configuration can reduce network load dramatically.

The following sections explain the basic components of a replication system, including replication objects, replication groups, and replication sites.

1.1 Replication Objects

A replication object is a database object existing on all servers in a replication system. In a replication environment, any updates made to a replication object at one site are applied to all other sites. Advanced Replication enables the replication of the following types of objects: Tables, Views, Indexes, Packages, Procedures and Functions, Triggers, User-Defined Types and Type Bodies.

1.2 Replication Groups

A replication group is a collection of replication objects that are logically related. By organizing related database objects within a replication group, it is easier to administer many objects together. Typically, a replication group is created and used to organize the schema objects necessary to support a particular database application.

1.3 Types of Replication Environments

Advanced Replication supports the following types of replication environments:

- Multimaster Replication
- Materialized View Replication
- Multimaster and Materialized View Hybrid Configurations

1.3.1 Multimaster Replication

Multimaster replication enables multiple sites, acting as equal peers, to manage groups of replicated database objects. Each site in a multimaster replication environment is a master site, and each site communicates with the other master sites, figure 1.

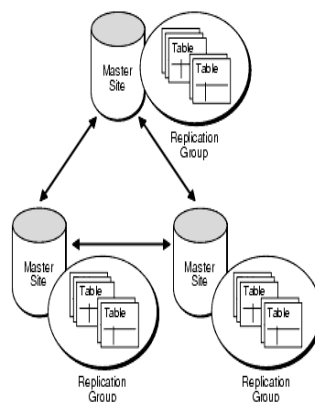


Fig. 1. Multimaster replication

During administrative tasks on the master group we must stop all replication activity. For example, all replication activity for a master group should be stopped to add a new master group object. Stopping all replication activity for a master group is called quiescing the group. When a master group is quiesced, users cannot issue DML statements on any of the objects in the master group. However, users can continue to query the tables in a quiesced master group.

1.3.2 Materialized View Replication

A materialized view takes a different approach in which the query result is cached as a concrete table that may be updated from the original base tables from time to time. This enables much more efficient access, at the cost of some data being potentially out-of-date, figure 2.

There are 3 types of Materialized Views:

1. Read-only views

- Cannot be updated and complex materialized views are supported

2. Updatable views

- Can be updated even when disconnected from the master site.
- Are refreshed on demand.
- Consume fewer resources.
- Require the Advanced Replication option to be installed.

3. Writable views

- Created with the FOR UPDATE clause.
- Changes are lost when views are refreshed.
- Requires Advanced Replication option to be installed.

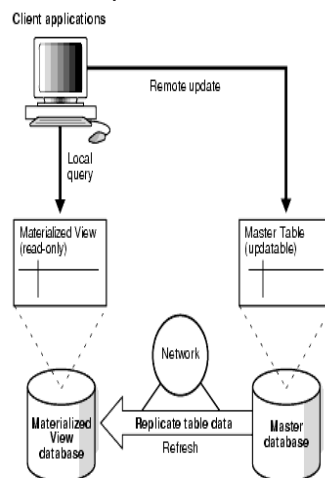


Fig. 2. Materialized View

1.3.3 Multimaster and Materialized View Hybrid Configurations

Multimaster replication and materialized views can be combined in hybrid or “mixed” configurations to meet different application requirements. Hybrid configurations can have any number of master sites and multiple Materialized View sites for each master, figure 3.

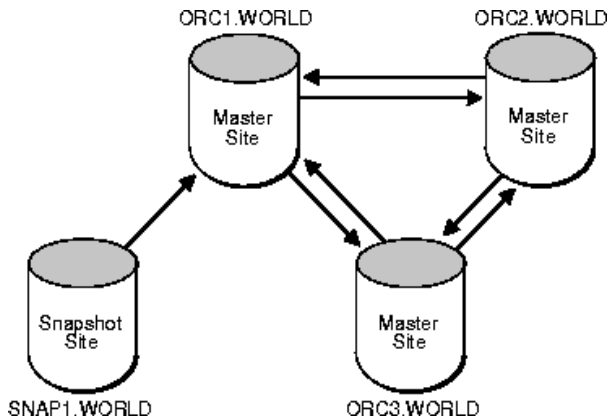


Fig. 3. Hybrid Configurations.

1.4 Why Use Multimaster Replication?

The following explain the reasons why Multimaster Replication, instead of Materialized View Replication should be used:

- In the Materialized View Replication there is one master site so all data is stored first in the master site and then the master site forwards all DML statements to all sites. If the master site goes down for some reason, the replication activity is automatically stopped.
- In a Materialized View Replication there is more network traffic. This problem does not exist in the Multimaster Replication.
- Another problem is that in Materialized View Replication, during the data refresh from master sites, users connected to the site database cannot make queries or execute DML statements.

2. Administrative tasks in multimaster replication

With the development of large software the probability of bugs in the software increases. In order to provide patches, updates to the objects should be prepared. It is possible that these updates should be made to replication objects bringing the need of doing some administrative tasks on the replication group. During administrative tasks the replication activity for the related replication group must be stopped. While administrative tasks are performed users cannot issue DML statements on any object

of the replication group. In order to add a new object or update an existing one, 3 to 4 minutes are needed to stop the replication activity. This amount of time is not a problem because usually the interval between two data refreshes is more than 5 minutes.

2.1 Adding a new site in the distributed system

Adding a new site in the distributed system is another administrative task that should be considered. In this process the replication activity must be stopped, replication objects in the new site must be created, the existing data should be copied on the new database server and finally the restart of the replication can be performed. This means that during this task the system is down. The following sections explain two different ways of adding a new site.

2.1.1 Adding New Master Sites with Online Instantiation

This is the standard method of adding a new site. But for large systems this method cannot be implemented because the replication activity must be stopped for hours. During tests performed in a system with 250 tables and about 35 million records with a communication rate of 1 Mbit/s between sites, the time amount consumed by this method is too long (about 30 hours). The replication activity cannot be stopped for 30 hours on systems that should be working 24x7.

The procedure of adding a new site with Online Instantiation follows:

```
BEGIN
  DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
    gname => 'replication_group');
END;
BEGIN
  DBMS_REPCAT.ADD_MASTER_DATABASE (
    gname => 'replication_group',
    master => 'dbx.rep',
    use_existing_objects => FALSE,
    copy_rows => TRUE,
    propagation_mode => 'ASYNCHRONOUS');
END;
BEGIN
  DBMS_REPCAT.RESUME_MASTER_ACTIVITY (
    gname => 'replication_group');
END;
```

2.1.2 Adding New Master Sites with Offline Instantiation

In order to minimize the time of the task of adding a new Master Site, Oracle has built another method named Offline Instantiation. The same logic applies in adding new Master Sites with Offline Instantiation as for the Online Instantiation, but this method

reduces the amount of time needed by eliminating the process of copying data when the replication activity is stopped. The solution is provided by exporting the tables of the replication group which can be imported only after the replication activity is restarted. In conclusion this method eliminates the time overhead of transferring data, in-serting and the building of indexes. The tests performed in the same conditions as for the Online Instantiation show a dramatic reduction of the amount of time needed (about 150 minutes). However, this amount of time is still a big problem for systems that are supposed to work 24x7.

The procedure of adding a new site with Offline Instantiation follows:

Step 1: Stopping the replication activity

```
BEGIN
  DBMS_REPCAT.SUSPEND_MASTER_ACTIVITY (
    gname => 'replication_group');
END;
```

Step 2: Beginning Instantiation

```
BEGIN
  DBMS_OFFLINE_OG.BEGIN_INSTANTIATION (
    gname => 'replication_group',
    new_site => 'dbx.rep');
END;
```

Step 3: Exporting data

```
exp system/manager SCHEMAS=rep_test DUMPFILE=rep_test_schema.dmp
```

Step 4: Restarting replication

```
BEGIN
  DBMS_OFFLINE_OG.RESUME_SUBSET_OF_MASTERS (
    gname 'replication_group', new_site => 'dbx.rep');
END;
```

Step 5: Importing data to the new site

```
BEGIN
  DBMS_OFFLINE_OG.BEGIN_LOAD (
    gname 'replication_group',
    new_site => 'dbx.rep');
END;
```

```
imp system/manager SCHEMAS=rep_test DUMPFILE= rep_test_schema.dmp
```

```
BEGIN
  DBMS_OFFLINE_OG.END_LOAD (
    gname 'replication_group',
    new_site => 'dbx.rep');
END;
```

Step 6: End procedure

```
BEGIN
  DBMS_OFFLINE_OG.END_INSTANTIATION (
    gname 'replication_group',
    new_site => 'dbx.rep');
END;
```

3. Minimization of the time needed for adding a new site

To minimize the “system down” time during the adding of a new Master Site in a Multimaster Replication System we propose a new method that uses the possibility to create more than one replication group and to make queries when the replication activity is stopped. We separate the replication group in more than one group. This separation process is based on the hierarchy of the replication objects and the logical relation between them.

Our proposal includes four replication groups:

- Base data replication group (Hierarchy level 1 / Highest level)
This group contains all base data tables. These tables usually are used only for queries. DML statements on them are very rare.
- Rare DML statements replication group (Hierarchy level 2)
This is a replication group with tables where DML statements happen rarely. This group contains non base data tables but only tables where DML statements happen not often.
- Main replication group (Hierarchy level 3)
This group contains tables where DML statements happen continually.
- Statistics replication group (Hierarchy level 4)
This group contains tables where we store data once per day or hour for statistics only.

3.1 Why do we separate replication objects in four groups?

If we separate tables as we said, we can stop the replication activity on one group and other groups can continue working with DML statements. In any case we can make queries to any table belonging to the stopped replication group. This way, when we try adding a new Master Site, the other sites do not stop working because they can make queries on this group.

Tables in a replication group must be organized according to the dependences between them. For example, in the Rare DML statements replication group, a child table (table that contains foreign keys) cannot be included if the parent table belongs to a lower group in the hierarchy, because during the import of data to the new site “parent key not found” errors would be generated.

The rest of the process which consists in adding a new server procedure is the same as for the Offline Instantiation. This causes the procedure to be repeated four times.

After various tests the “system down” time resulted to be about 20 minutes. In this way the most critical factor affecting the availability of the system has been dramatically reduced. This method has been successfully implemented in a real world system, working 24x7.

Conclusions

- The method simplifies the maintenance of Multimaster Replication systems because in order to change a replication object, stopping the entire replication system is not necessary. With this method, only the stopping of the replication group, which contains this replication object, is needed.
- Adding a new server is more complicated but the system doesn’t stop working for a long time.
- In order to remove a replication group from a Multimaster replication system there is no need to destroy all replication groups, but only that group.

Bibliography

1. CICO.B,PACI.H, Oracle Multimaster Data Replication Optimization, International Scientific Conference Computer Science, 2008
2. Burleson, D.K., Garmany, J., Karam, S. (2003) Oracle Replication, Rampant.
3. Differences Between Read-Only, Updatable and Writeable Materialized Views’ Metalink article, 162711.1
4. Dye, Ch. (1999) Oracle Distributed Systems, O’Reilly
5. Oracle Master Replication, Performance and Scalability, Metalink article 76448.1
6. Tumma, M., (2004) Oracle Streams, High Speed Replication and Data Shar-ing, Rampant.