

Bidirectional Replication (Part I)

by Ahmed Baraka

Introduction to Oracle Data Guard

In this lecture, we are going to talk about the basic concepts of Oracle Data Guard

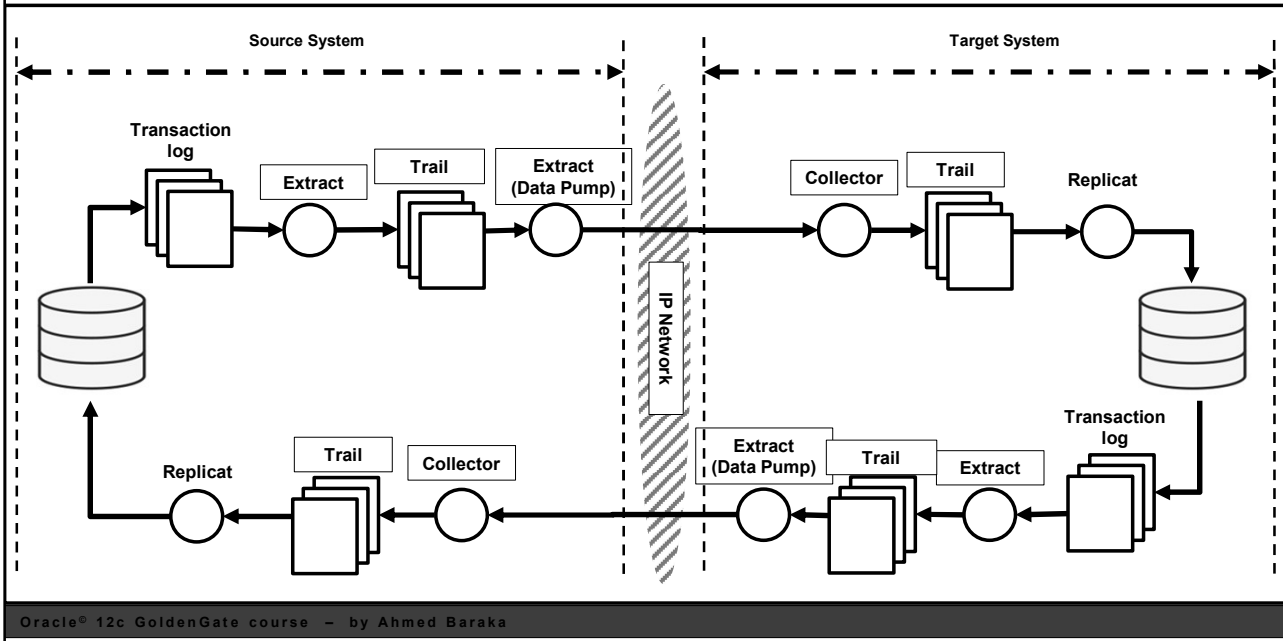
Objectives



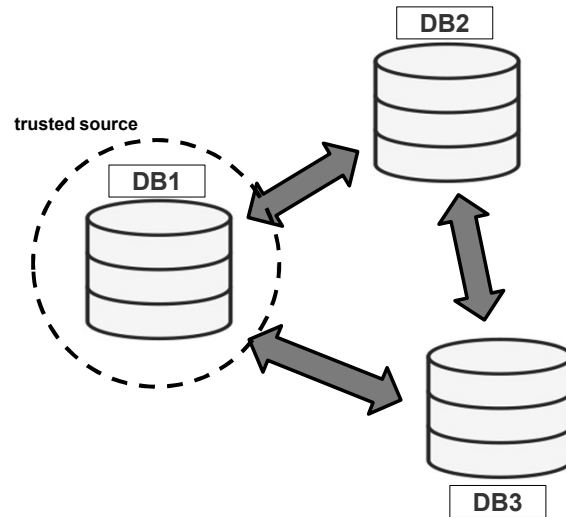
By the end of this lecture, you should be able to:

- Understand how the bidirectional replication works in Oracle GoldenGate
- Understand the challenges in implementing a bidirectional configuration
- Understand the Data Looping issue and how to prevent it

Oracle GoldenGate Architecture in Bidirectional Configuration



Oracle GoldenGate Multi-master Topology



About Bidirectional Configuration



- Supported in heterogeneous configurations
- Active-active configuration
- Supported on: Oracle, SQL Server, MySQL, SQL/MX, Sybase, Teradata, and DB2 on z/OS, LUW, and IBM I
- Enables increased availability and performance
- Continuous operations, additional raw computing capacity, and the flexibility to optimize workload management

Bidirectional Configuration Challenges



- Application Design
- Sequence numbers and identity data types
- Truncate table operations
- Triggers and Cascaded Deletes
- Loop detection
- Conflict prevention, detection, and resolution

Sequence Numbers and Identity Data Types

- It is highly recommended to have a primary key or unique identifier for all the data tables
- The primary key generation mechanism must maintain the uniqueness in all the databases



Ahmed Baraka
Oracle Database Administrator

Truncate Table Operations



- Truncate table can be supported in one direction
- Configure the TRUNCATE parameter in one database
- Always truncate the table from the same database
- On that database, configure the Extract and Replicate to have the GETTRUNCATES parameter
- Disallow the truncate from the other databases
- On those other databases, configure the Extract and Replicat parameter files to contain IGNORETRUNCATES parameter

Handling Triggers in Bidirectional Configuration



- Modify triggers to ignore DML operations that are applied by Replicat (not needed in case of Integrated Replicat)

```
CREATE TRIGGER .. BEFORE UPDATE
ON ..
FOR EACH ROW
BEGIN
IF SYS_CONTEXT ('USERENV', 'SESSION_USER') != 'OGGS' THEN
    ...
END IF;
...
```

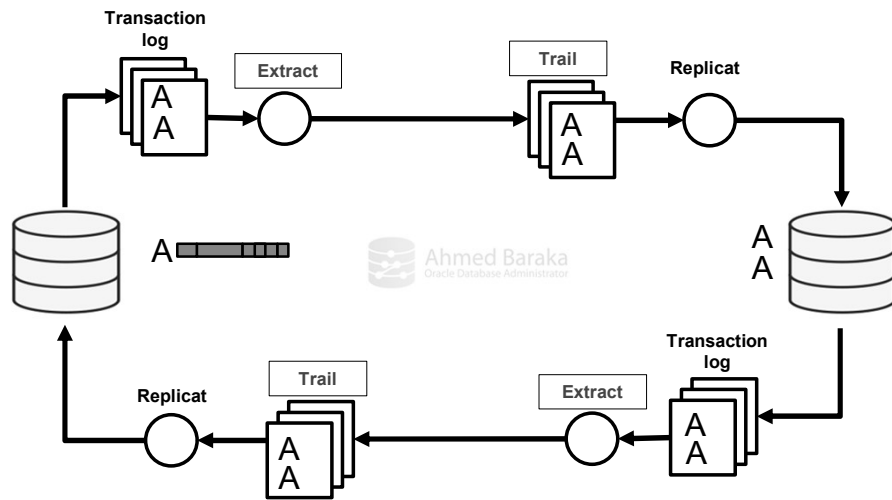
Cascaded Deletes in Bidirectional Configuration



Ahmed Baraka
Oracle Database Administrator

- Disable ON DELETE CASCADE constraints and use a trigger on the parent table to perform the required delete(s) to the child tables.
- Integrated Replicat automatically handles them.

About Data Looping



Oracle® 12c GoldenGate course – by Ahmed Baraka

Preventing Data Looping

- Problem: all transactions will loop endlessly
- Solution: do not capture Replicat transactions



Preventing Data Looping Options



- Capture application DMLs and ignore Replicat transactions:

```
GETAPPLOPS and IGNOREREPLICATES
```

- Specify the name of the Replicat database user to be used by GETREPLICATES or IGNOREREPLICATES:

```
TRANLOGOPTIONS ...[EXCLUDEUSER <user>]
```

```
TRANLOGOPTIONS ...[EXCLUDEUSERID <oracle_uid>]
```

- Ignore transaction with the specified redo tag (Oracle 12c):

```
TRANLOGOPTIONS ...[EXCLUDETAG [<tag>]]
```

Preventing Data Looping Options (cont)



- Exclude Replicat transaction name:

```
TRANLOGOPTIONS ...[EXCLUDETRANS <transaction>]
```

- Store Replicat transaction information in a table:

- Use an Oracle trace table

```
GGSCI> ADD TRACETABLE
```

- Identify the name of the Replicat checkpoint table in the TRANLOGOPTIONS (MySQL and NonStop SQL/MX)

```
TRANLOGOPTIONS FILTERTABLE <table_name>
```

Preventing Data Looping: Ignore Replicat database user transactions



Ahmed Baraka
Oracle Database Administrator

```
GETAPPLOPS  
IGNOREREPLICATES  
TRANLOGOPTIONS EXCLUDEUSER ggsrep
```

```
GETAPPLOPS  
IGNOREREPLICATES  
TRANLOGOPTIONS EXCLUDEUSERID 90
```

Preventing Data Looping: Ignore transaction with the specified redo tag



- Replicat parameter file:

```
DBOPTIONS SETTAG 0935
```

- Extract parameter file:

```
TRANLOGOPTIONS EXCLUDETAG 0935
```


Preventing Data Looping: Exclude Replicat transaction name



- Extract parameter file:

```
TRANLOGOPTIONS EXCLUDETRANS ggs_rep1
```

Preventing Data Looping: Use an Oracle trace table



- Add a trace table:

```
GGSCI> ADD TRACETABLE [[<container>.<owner.table>]
```

- If you are not using the default table name, add a parameter in both Extract and Replicat:

```
TraceTable <owner>.<table_name>
```

Preventing Data Looping: Identify the name of the Replicat checkpoint table



- A Replicat checkpoint must be there:

```
GGSCI> ADD CHECKPOINTTABLE ggs.fin_check
```

- Instruct the Extract to ignore transactions that contain any operation on the checkpoint table:

```
TRANLOGOPTIONS FILTERTABLE ggs.fin_check
```

Preventing Data Looping Options

| Recommended Option/DB | Oracle | SQL Server | DB2 | MySQL | Sybase | Teradata |
|-----------------------|--------|------------|-----|-------|--------|----------|
| GETAPPLOPS | X | X | X | X | X | X |
| IGNOREREPLICATES | | | | | | |
| .. EXCLUDEUSER | X | | X | | X | |
| .. EXCLUDEUSERID | X | | | | | |
| EXCLUDETAG | X | | | | | |
| .. EXCLUDETRANS | | X | | | X | |
| Trace File | X | | | | | |
| Checkpoint Table | | | | X | | |

Summary



In this lecture, you should have learnt the following:

- Understand how the bi-directional replication works in Oracle GoldenGate
- Understand the challenges in implementing a bidirectional configuration
- Understand the Data Looping issue and how to prevent it