

GOLDENGATE REPLICATION CONFLICT DETECTION AND RESOLUTION AND ITS CHALLENGES

My GoldenGate Journey

- Supported 2-way and 3-way replication with legacy Advanced Replication
- Started with GoldenGate 10
- Supported GoldenGate 11.1
- Supporting GoldenGate 11.2 and 12.1 one way, 2 way and 3+ way replication.
- Supporting combination of GoldenGate and standby.

Agenda

- GoldenGate usage, is it a good fit your app?
- GoldenGate requirements
- GoldenGate basic concepts
- GoldenGate conflict detection and resolution
- Other GoldenGate advanced setup
- Operational challenges of multi-way replication

AUDIENCE SURVEY



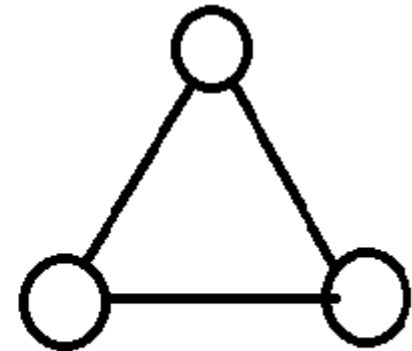
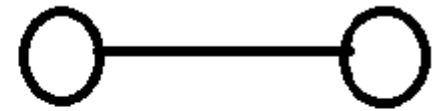
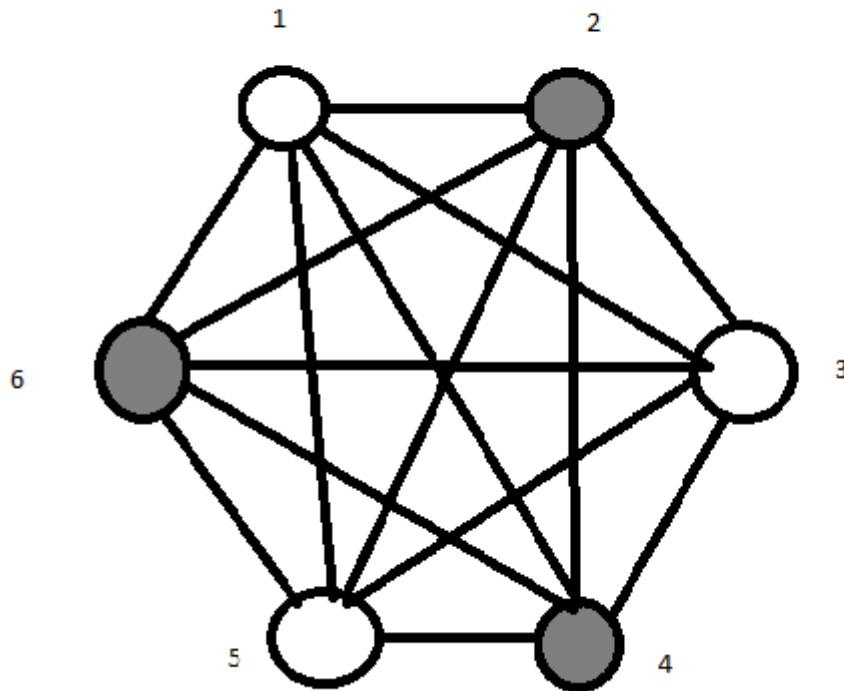
GoldenGate Usage

- One way replication
 - ▣ One time data migration across DB platforms
 - ▣ Ongoing master to slave database copies for business or technical reasons
 - ▣ GoldenGate vs. DataGuard
- Two way replication
 - ▣ For data redundancy as well as load sharing.
- Three way plus/multi-way replication
 - ▣ Complexity
 - ▣ Workload increases exponentially (next page)
- Combination of GoldenGate and Data Guard
 - ▣ GoldenGate active on primary database but dormant on standby

Multi-way replication workload comparison in simplest case

N-way	# of extract at each site	# of pumps at each site	# of replicats at each site	Total # of processes at each site	Total # of processes for all sites
2	1	1	1	3	6
3	1	2	2	5	15
4	1	3	3	7	28
5	1	4	4	9	45
6	1	5	5	11	66
N	1	N-1	N-1	2N-1	$N*(2N-1)$

Visual comparison of 2, 3 and 6 way





**GoldenGate is NOT application
transparent !!!**

Requirements on application

- Each replicated table must have a primary key, or unique key, or a set of columns (specified by KEYCOLS in extract parameter file) to uniquely identify a row. (Referred as replication key for simplicity in this presentation.)
- Restriction on some data types.
- In a multi-master setup, each replicated table should have conflict rules defined and configured. New column may need be added for conflict resolution purpose.
 - ▣ Common practice
 - Artificial primary key column populated by a sequence
 - Last updated timestamp column or and/or site column
- DDL replication consideration.
- Monitor replication data exceptions and perform manual data fix when data is out of sync
- Have a process to check data is in sync on an on-going basis.
- There can be further requirements on application code due to business need (at end of this presentation)

Requirements on DBAs

- Install GoldenGate software.
- Make necessary database parameter and configuration changes for GoldenGate. For example,
 - ▣ ALTER SYSTEM SET ENABLE_GOLDENGATE_REPLICATION=true;
- Create necessary database users to be used by GoldenGate extract and/or replicat.
- Create GoldenGate parameter files for extract, pipe and replicat
- Create extract, pipe, replicat
- Install XAG software and configure it to automatically failover GoldenGate if database is RAC.
- GoldenGate error monitoring and performance tuning

Basic Concepts

- Manager
- Extract – the data change grabber at source database
- Pump – the shipper
- Replicat – the change applier at target database
- Trail files consumed by pump(s) or by replicat(s)
EXTTRAIL <directory>/file_prefix
RMTTRAIL <directory>/file_prefix
- Good practice - Have a good name convention

```
GGSCI 1> info all
```

```
Program      Status      Group      Lag at Chkpt  Time Since Chkpt
```

```
MANAGER      RUNNING
```

```
EXTRACT      RUNNING    E1xxxxxx  00:00:09    00:00:01
```

```
EXTRACT      RUNNING    P2xxxxxx  00:00:00    00:00:02
```

```
REPLICAT     RUNNING    R2xxxxxx  00:00:00    00:00:05
```

Advanced Configurations

- Conflict detection and resolution (today's subject)
- Source and target tables have different definitions
- Data filtering and manipulation
- Performance tuning

Conflict Detection and Resolution

- Majorly for 2 way replication and above
- Conflict
 - how is it detected
 - Types
 - Set up
- Resolution
 - built-in
 - customized

Conflict Detection

- What is considered a data conflict?

A conflict is a mismatch between the before image of a record in the trail and the current record in the target table.

- How is current record on the target table found?

By replication key

Conflict Type

- Insert conflict
Uniqueness violation on replication key, primary key or unique key
- Update row missing
The row can not be found by the replication key.
- Update row exists
The row is found by replication key but before image of incoming record doesn't match current record on compared columns in target database.
- Delete row missing
The row can not be found by the replication key.
- Delete row exists
The row is found by replication key but before image of incoming record doesn't match current record on compared columns in target database.

Enable Conflict Detection

- Specify the columns that replicats use to detect update and delete conflict in replicat file

MAP scott.table1, TARGET scott.table1, &

COMPARECOLS (ON UPDATE <columns>, ON DELETE <columns>)

- Different ways to specify <columns>
 - ▣ ALL (highest workload, best data convergence)
 - ▣ KEY (replication key)(fastest, least data quality)
 - ▣ KEYINCLUDING (col1, ...)
 - ▣ ALLEXCLUDING (col1, ...)
 - ▣ KEYANDMOD (key and modified columns)

COMPARECOLS requires that

- The before image must be present in the trail file
- Specified columns must exist in the target database
- Can only compare scalar data types
 - ▣ Scalar data types
Numeric, Date, Character
 - ▣ Non Scalar data columns must be excluded from the comparison. For example, LOBs, user defined, spatial, reference, raw etc.

Put before image into DB log files

- DB default logging is not enough
- Enable DB level force logging and minimum supplemental logging
 - ALTER DATABASE FORCE LOGGING; -- force logging of all transactions
 - ALTER DATABASE ADD SUPPLEMENTAL LOG DATA; --add row chaining info into log
 - ALTER SYSTEM SWITCH LOGFILE;
- Enable schema or table level supplemental logging
 - ADD SCHEMATRANDATA (when DDL replication is included)
 - ADD TRANDATA
 - By default, log primary key unconditionally weather the key is changed or not and log scheduling columns (primary key, unique key, and foreign key columns) if one of them is changed.
 - NOSCHEDULINGCOLS – unconditionally log only PK and UK.
 - **ALLCOLS** - unconditional supplemental logging of all supported columns
 - COLS (col1, col2) – log listed columns
 - NOKEY – used in conjunction with COLS.

Effect of ADD TRANDATA scott.table1 ALLCOLS

```
SQL> select * from dba_log_groups
      where table_name='TABLE1';
```

OWNER	LOG_GROUP_NAME		
TABLE_NAME	LOG_GROUP_TYPE	ALWAYS	GENERATED
SCOTT	GG5_18413		
TABLE1	USER LOG GROUP	ALWAYS	USER NAME
SCOTT	SYS_C006351		
TABLE1	PRIMARY KEY LOGGING	ALWAYS	GENERATED NAME
SCOTT	SYS_C006352		
TABLE1	UNIQUE KEY LOGGING	CONDITIONAL	GENERATED NAME
SCOTT	SYS_C006353		
TABLE1	FOREIGN KEY LOGGING	CONDITIONAL	GENERATED NAME
SCOTT	SYS_C006354		
TABLE1	ALL COLUMN LOGGING	ALWAYS	GENERATED NAME

Put before image into GG trail file

- In extract parameter file
 - All columns will be captured when there is no PK, UK column(s), or alternate key specified by TABLE ...KEYCOLS (columns)
 - LOGALLSUPCOLS in GG12.
 - Record the before image of all supplemental logged columns for both UPDATE and DELETE operations.
 - NOLOGALLSUPCOLS is default.
 - GETUPDATEBEFORES and NOCOMPRESSUPDATES below GG12
 - IGNOREUPDATEBEFORES and COMPRESSUPDATES are default
 - TABLE scott.table1 COLS (col1, col2), must include key columns in the list
 - TABLE scott.table1 COLSEXCEPT (col1, col2), does not exclude key columns
 - TABLE scott.table1, GETBEFORECOLS (ON UPDATE <option>, ON DELETE <option>); <options> are
 - ALL – all supported columns (highest workload)
 - KEY – this the default.
 - KEYINCLUDING (col1, col2)
 - KEYANDMOD – include modified columns
 - ALLEXCLUDING (col1, col2)

Conflict Resolution

Out of the box

Conflict type	Conflict detail	Resolution available
Insert conflict on replication key (unique constraint conflict)	Insert row exists	Overwrite Ignore Discard USEMIN, USEMAX
Update conflict 1	Update row missing (The row can not be found by the replication key.)	Overwrite Ignore Discard
Update conflict 2	Update row exists (The row is found by replication key but before image of incoming record doesn't match current record on compared columns in target database.)	Overwrite Ignore Discard USEMIN, USEMAX USEDELTA
Delete conflict 1	Delete row missing (The row can not be found by the replication key.)	Ignore Discard
Delete conflict 2	Delete row exists (The row is found by replication key but before image of incoming record doesn't match current record on compared columns in target database.)	Overwrite Ignore Discard (note that overwrite will require before image of all columns be sent through thus not suitable for tables with many columns)

Difference between Discard and Ignore

- Discard

Retain the current value in the target database, and write the data in the trail record to the discard file.

- Ignore

Retain the current value in the target database, and ignore the trail record. No record is written into discard file.

Conflict Resolution out of the box

- Example

```
MAP scott.table1, TARGET scott.table1, &  
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &  
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT,IGNORE)), &  
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD)),  
&  
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, IGNORE)), &  
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT,  
OVERWRITE)), &  
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX  
(last_updated)) );
```

Conflict Resolution

customized solutions

- When built-in resolution can not satisfy your need.
- Uniqueness conflict on non primary key column as an example.

Conflict Resolution

other considerations

- Adjust sequences used to populate primary or unique keys
- Using triggers to populate resolution based columns such as last updated timestamp is preferred,
- Use same database/host time zone for replicated databases if timestamp based resolution is chosen.
- Group tables with parent child relationship, or tables that can have logical relationship in one transaction in same replication pipe
- Exception table or discard files
 - Discard file
DISCARDFILE <directory>/replicat.dsc, MEGABYTES 50, APPEND
DISCARDROLLOVER at 12:00
 - Exception table
MAP scott.table1, TARGET scott. gg_exception, EXCEPTIONSONLY, INSERTALLRECORDS & COLMAP (...);

Conflict Resolution

other considerations

- REPEROR (error, response) in replicate file
Can be global or at each table level (in MAP statement)

```
REPEROR (  
  DEFAULT | DEFAULT2 | SQL error |  
  ABEND                - default behavior  
  IGNORE               - ignore the error.  
  DISCARD              - send to discard file  
  EXCEPTION            - to be handled by MAP exception statement  
  TRANSABORT           - abort at transaction level  
  TRANSDISCARD  
  TRANSEXCEPTION  
  RETRYOP [ MAXRETRIES n] – useful for transit error  
)
```

Operational challenges

- For a multi-master replicated environment that doesn't allow application downtime
 - ▣ Application release must be backward/forward compatible with DB code (tables, packages.)
 - ▣ May have to complete all database table changes in all replicated databases before/after application code change.
 - ▣ Column addition/deletion on replicated tables



Q & A