Monitoring Oracle GoldenGate

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:

- Display the status of the processes
- Obtain reports on the Extract and Replicat lags
- Implement Automatic Heartbeat Tables functionality
- View and Understand the process reports and discard files

Process Status

Status	Description
STARTING	The process has started but has not yet locked the checkpoint file for processing
RUNNING	The process is Active or Suspended
STOPPED	The process was stopped
ABENDED	The process abnormally ended

Obtaining Process Information

```
Info All
Info Manager | Mgr
Info {Extract | Replicat} <group> [Detail]
Stats {Extract | Replicat} <group>
Status {Extract | Replicat} <group>
Status Manager
Lag {Extract | Replicat} <group>
```

Obtaining Process Information (cont)

```
Info {ExtTrail | RmtTrail} <path_name>
Send Manager
Send {Extract | Replicat}
View Report <group>
View GGSEvt
```

Using `Info Extract` or `Replicat`

```
INFO EXTRACT | REPLICAT group_name
    [, SHOWCH [n]]
    [, DETAIL]
```

```
INFO EXTRACT ehr1, DETAIL
```

```
INFO EXTRACT ehr1, SHOWCH
```

Monitoring Lag

- **Extract lag**: the difference, in seconds, between the time that a record was processed by Extract and the timestamp of that record in the data source.

- **Replicat Lag**: the difference, in seconds, between the time that the last record was processed by Replicat and the timestamp of that record in the trail.

```
LAG EXTRACT E*
LAG REPLICAT R*
```

Controlling How Lag is Reported

Parameters	Description
`LAGREPORTMINUTES, LAGREPORTHOURS`	The interval at which the lag will be measured
`LAGINFOSECONDS, LAGINFOMINUTES, LAGINFOHOURS`	A lag info threshold: if lag exceeds the specified value, lag information will be reported in the error log file as *information*.
`LAGCRITICALSECONDS, LAGCRITICALMINUTES, LAGCRITICALHOURS`	A lag critical threshold: if lag exceeds the specified value, lag information will be reported in the error log file as *critical*.

Monitoring Lag with Automatic Heartbeat Tables

- Best for measuring end-to-end replication lag
- It should be implemented manually before 12.2
- Functionality components: three tables, two views, two procedures, and two scheduler jobs
- Information saved: source and target databases, all the processes included in the incoming and outgoing streams with processing timestamp
- Clocks must be set correctly on the systems

Implementing Automatic Heartbeat Tables

1. Ensure that Self-Describing Trail Files functionality is enabled
2. Login to the database:

```
GGSCI>DBLOGIN USERID <username>, PASSWORD <password>
```

3. Add the heartbeat table functionality components:

```
GGSCI>ADD HEARTBEATTABLE
GGSCI>ADD HEARTBEATTABLE, frequency 120,
retention_time 10, purge_frequency 2
```

4. Repeat for target database
5. Restart the processes

Automatic Heartbeat Tables Components

Component	Type	Description
GG_HEARTBEAT_SEED	Table	Gets updated automatically by the job
GG_HEARTBEAT	Table	Gets updated automatically by the Replicat
GG_HEARTBEAT_HISTORY	Table	History of the rows updated on the heartbeat tables
GG_UPDATE_HB_TAB	Proc.	A procedure to update the heartbeat tables
GG_PURGE_HB_TAB	Proc.	A procedure to purge the heartbeat table contents
GG_UPDATE_HEARTBEATS	Job	A job to call the update procedure
GG_PURGE_HEARTBEATS	Job	A job to call the purge procedure
GG_LAG_HISTORY	View	View based on the GG_HEARTBEAT_HISTORY

Display the Heartbeat Data

- To monitor the lag through the heartbeat tables:

```
SELECT HEARTBEAT_RECEIVED_TS, INCOMING_PATH,
  INCOMING_LAG FROM GG_LAG_HISTORY;
```

- LAG command accesses the view `GG_LAG` and `GG_LAG_HISTORY`

```
DBLOGIN USERID <dbusername>, PASSWORD <password>
LAG rhr1
```

Manage Heartbeat Functionality

Command	Description
`INFO HEARTBEATTABLE`	Displays heartbeat table configuration information
`ADD HEARTBEATTABLE`	Creates the objects required for automatic heartbeat functionality.
`ALTER HEARTBEATTABLE`	Alters existing heartbeat configuration
`DELETE HEARTBEATTABLE`	Deletes existing heartbeat objects
`DELETE HEARTBEATENTRY`	Deletes entries in the heartbeat table.

Process Report Files

- Information included:
  - Parameters in use
  - Table and column mapping
  - Database information
  - Runtime messages and errors
  - Runtime statistics

- Can be viewed with:

```
GGSCI> View Report <group name> | Mgr
```

Discard Files

- Failed GoldenGate data operations.
- Information may include:
  - database error message
  - the sequence number of the data source or trail file
  - the relative byte address of the record in the data source or trail file
  - the details of the discarded operation, such as the column values of a DML statement or the text of a DDL statement

Maintaining the Discard and Report Files

- A new discard or report file is created at the start of a new process run.
- If file reaches maximum size, the process by default abends.
- Set a schedule for aging report and discard files
  - **REPORTROLLOVER**
  - **DISCARDROLLOVER**

```
DiscardRollover at 02:00 ON SUNDAY
```

Using the ggserr.log Error Log

- Error log provides the following:
  - Errors that occurred
  - History of GGSCI commands
  - The processes that started and stopped
  - Informational and warning messages

- Can be viewed by an editor or:

```
View GGSEVT
```

Using the System Logs

- Errors are written in OS error logs:
  - Event Viewer in Windows (DLL must be registered)
  - **`syslog`** in Linux

- **`SYSLOG`** parameter controls the types of messages to log:
  - include all Oracle GoldenGate messages
  - suppress all Oracle GoldenGate messages
  - filter to include information, warning, or error messages, or any combination of those types

Summary

In this lecture, you should be able to:

- Display the status of the processes
- Obtain reports on the Extract and Replicat lags
- Implement Automatic Heartbeat Tables functionality
- View and Understand the process reports and discard files