

Operaciones DML sobre Active Dataguard - Oracle Database 19c

Por Francisco Riccio

Introducción

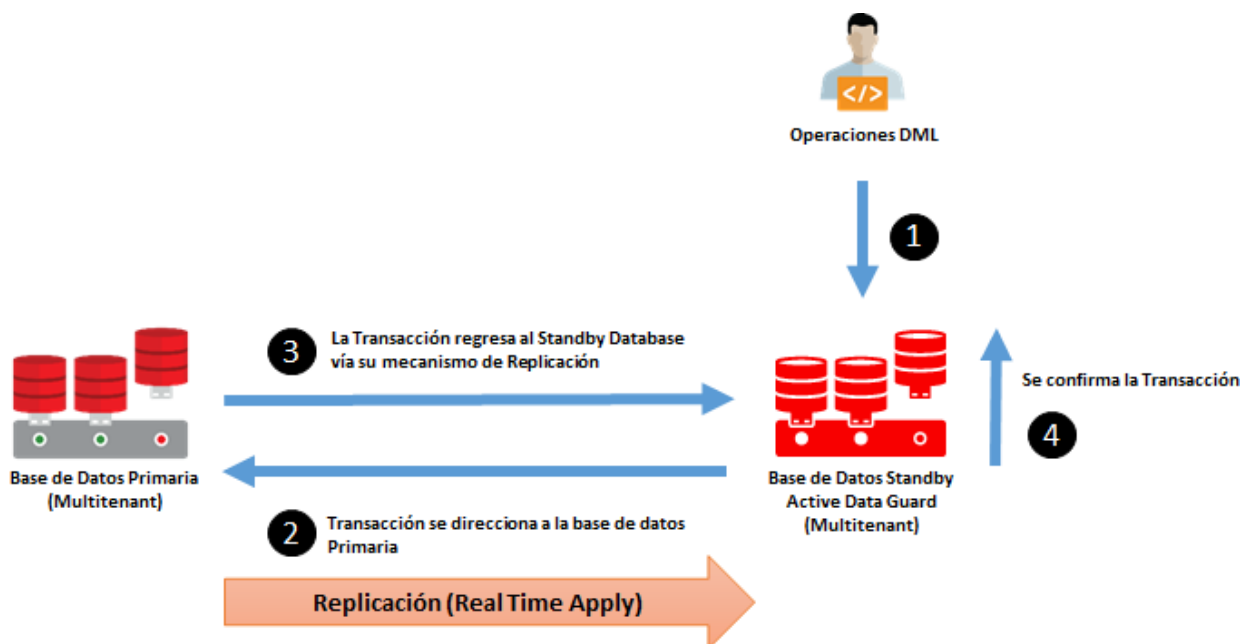
Uno de los principales beneficios que nos provee Active Dataguard (disponible desde Oracle Database 11g) es mantener una base de datos de contingencia pero con la posibilidad de realizar consultas, donde los datos leídos son actualizados vía archived logs o a nivel de transacción en línea (Real Time Apply, provisto en Oracle Database 10gR2).

Oracle Database 12cR1 extendió la funcionalidad permitiendo crear tablas temporales sobre las instancias de base de datos en contingencia con la finalidad de ejecutar reportes que requieran escribir temporalmente información como parte de su proceso.

Todas estas mejoras fueron incluidas para aprovechar las capacidades del ambiente de contingencia y liberar carga al ambiente productivo; pero una de las necesidades que muchos Administradores de Base de Datos anhelaban es permitir realizar operaciones DML sobre la base de datos Standby y ahora esto es posible en la versión Oracle Database 19c.

La base de datos logra este objetivo re-direccionando las transacciones ejecutadas en el Standby hacia la base de datos Primaria provocando que regrese luego la información vía sus mecanismos de replicación.

A continuación se adjunta los pasos que ocurren cuando se ejecuta una operación DML sobre la base de datos de Active Data Guard:



Como se puede apreciar la base de datos Standby re-direcciona la transacción hacia la base de datos Primaria y por dicha razón no es recomendable realizar gran cantidad de operaciones DML sino para transacciones puntuales como parte de un proceso, además tener presente que la latencia de red entre las 2 base de datos puede ser un factor determinante sobre el tiempo de ejecución de dichas operaciones.

Esta nueva funcionalidad soporta configuraciones Multitenant y Single Tenant como también para replicaciones vía Real Time Apply que es la opción recomendada o a través de archived logs; asimismo funciona sobre los 3 métodos de protección disponibles en un Standby.

Implementación

A. Revisión de las Configuraciones del Data Guard

A continuación se presentará las configuraciones de los ambientes:

ROL PRIMARIO	
IP	132.68.1.19
Modo de Replicación:	Real Time Apply
db_name	PRD
db_unique_name	PRD_01
log_archive_config	DG_CONFIG=(PRD_01,PRD_02)
log_archive_dest_1	LOCATION=USE_DB_RECOVERY_FILE_DEST
log_archive_dest_2	SERVICE=CTG ASYNC VALID_FOR=(O NLINE_LOGFILE,PRIMARY_ROLE) DB_UNIQUE_NAME=PRD_02
fal_server & fal_client	Valor vacío para ambos parámetros
local_listener	LISTENER_PRD

ROL STANDBY	
IP	132.68.1.20
Modo de Replicación:	Real Time Apply
db_name	PRD
db_unique_name	PRD_02
log_archive_config	DG_CONFIG=(PRD_01,PRD_02)
log_archive_dest_1	LOCATION=USE_DB_RECOVERY_FILE_DEST
log_archive_dest_2	SERVICE=PRD ASYNC VALID_FOR=(O NLINE_LOGFILE,PRIMARY_ROLE) DB_UNIQUE_NAME=PRD_01
fal_server & fal_client	Valor vacío para ambos parámetros

local_listener	LISTENER_CTG
----------------	--------------

Se adjunta la configuración del archivo TSNAMES.ORA el cual es el mismo para ambos servidores:

```
LISTENER_PRD =
  (ADDRESS = (PROTOCOL = TCP) (HOST = srvbd.riccio.com) (PORT = 1521))

LISTENER_CTG =
  (ADDRESS = (PROTOCOL = TCP) (HOST = srvbdctg.riccio.com) (PORT = 1521))

PDB_CTG =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 132.68.1.20) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = PDB_PRD)
    )
  )

PDB_PRD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 132.68.1.19) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = PDB_PRD)
    )
  )

PRD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = srvbd.riccio.com) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = PRD_01)
    )
  )

CTG =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = srvbdctg.riccio.com) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = PRD_02)
    )
  )
```

Configuración de los Standby Redo Logs:

```
SQL> select group#,type from v$logfile;
GROUP# TYPE
-----
1 ONLINE
2 ONLINE
3 ONLINE
4 STANDBY
5 STANDBY
6 STANDBY
7 STANDBY

7 rows selected.

SQL> select database_role from v$database;
DATABASE_ROLE
-----
PRIMARY

SQL> select group#,type from v$logfile;
GROUP# TYPE
-----
1 ONLINE
2 ONLINE
3 ONLINE
4 STANDBY
5 STANDBY
6 STANDBY
7 STANDBY

7 rows selected.

SQL> select database_role from v$database;
DATABASE_ROLE
-----
PHYSICAL STANDBY
```

Los standby redo logs son importantes porque permiten habilitar la funcionalidad de Real Time Apply consiguiendo replicar a nivel de transacción hacia el Standby Database, **siendo la mejor opción si queremos usar la nueva funcionalidad** a desarrollar en este material.

Recordar como buena práctica: # Grupos de Standby Redo Logs = # Grupos de Redo Logs + 1
Asimismo los tamaños de los grupos de los standby redo logs deben ser iguales a los redo logs.

Validación de la correcta configuración de Real Time Apply:

En la base de datos Primaria:

```
SQL> select dest_name,type,recovery_mode from v$archive_dest_status where dest_id=2;
DEST_NAME          TYPE          RECOVERY_MODE
-----
LOG_ARCHIVE_DEST_2  PHYSICAL     MANAGED REAL TIME APPLY
```

Se consulta el ID=2 porque es el canal asignado para el envío de los archived logs y transacciones al Standby en la configuración presentada.

En la base de datos Standby:

```
SQL> select dest_name,type,recovery_mode from v$archive_dest_status where dest_id=1;
DEST_NAME          TYPE          RECOVERY_MODE
-----
LOG_ARCHIVE_DEST_1  LOCAL        MANAGED REAL TIME APPLY
```

Finalmente el inicio de la réplica se ejecuta en la base de datos Standby a través del siguiente comando con la base de datos en OPEN READ ONLY:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE DISCONNECT FROM SESSION;
```

Nota: Se ha presentado las configuraciones del ambiente con la finalidad de entender el contexto más no de explicar cómo configurar un Oracle Data Guard, en caso se requiera revisar los pasos de implementación se recomienda revisar la siguiente documentación:

https://docs.oracle.com/database/121/SBYDB/create_ps.htm

B. Implementación de la Funcionalidad

A continuación se creará una tabla llamada Producto en un PDB llamado PDB_PRD.

```
[oracle@srvbd /]$ sqlplus friccio/oracle@PDB_PRD

SQL*Plus: Release 19.0.0.0.0 - Production on Sat Mar 30 19:08:06 2019
Version 19.2.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.2.0.0.0

SQL> select database_role from v$database;

DATABASE_ROLE
-----
PRIMARY

SQL> create table Producto (id number, nombre varchar(20), pu number);

Table created.

SQL> insert into Producto values (100,'Producto1',1000);

1 row created.

SQL> commit;

Commit complete.
```

En la base de datos de Active Data Guard la información ya se encuentra disponible:

```
[oracle@srvbdctg /]$ sqlplus friccio/oracle@PDB_CTG;

SQL*Plus: Release 19.0.0.0.0 - Production on Sat Mar 30 19:13:58 2019
Version 19.2.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

Last Successful login time: Sat Mar 30 2019 19:08:07 -05:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.2.0.0.0

SQL> select database_role, open_mode from v$database;

DATABASE_ROLE      OPEN_MODE
-----
PHYSICAL STANDBY READ ONLY WITH APPLY

SQL> select * from Producto;

          ID NOMBRE                PU
-----
100 Producto1                1000
```

Ahora en la base de datos Standby realizaremos una transacción, el cual puede ser configurado a nivel de sesión o a nivel de instancia.

A nivel de Sesión:

Se realiza a través del comando: ALTER SESSION ENABLE ADG_REDIRECT_DML

```
SQL> select database_role, open_mode from v$database;

DATABASE_ROLE      OPEN_MODE
-----
PHYSICAL STANDBY  READ ONLY WITH APPLY

SQL> select * from Producto;

      ID NOMBRE                PU
-----
      100 Producto1            1000

SQL> ALTER SESSION ENABLE ADG_REDIRECT_DML;

Session altered.

SQL> insert into Producto values (200,'Producto2',2000);

1 row created.

SQL> commit;

Commit complete.
```

En la base de datos Primaria se visualizará la fila ingresada en la base de datos Standby.

```
SQL> select database_role from v$database;

DATABASE_ROLE
-----
PRIMARY

SQL> select * from Producto;

      ID NOMBRE                PU
-----
      100 Producto1            1000
      200 Producto2            2000
```

A nivel de Instancia:

Es posible configurar esta opción a nivel de instancia a través del siguiente parámetro: `ADG_REDIRECT_DML=TRUE`.

Ejemplo:

```
[oracle@srvbdctg /]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Sat Mar 30 19:28:52 2019
Version 19.2.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.2.0.0.0

SQL> alter system set adg_redirect_dml=true;

System altered.

SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.2.0.0.0
[oracle@srvbdctg /]$ sqlplus friccio/oracle@PDB_CTG;

SQL*Plus: Release 19.0.0.0.0 - Production on Sat Mar 30 19:29:30 2019
Version 19.2.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

Last Successful login time: Sat Mar 30 2019 19:24:24 -05:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.2.0.0.0

SQL> select database_role,open_mode from v$database;

DATABASE_ROLE      OPEN_MODE
-----
PHYSICAL STANDBY  READ ONLY WITH APPLY

SQL> insert into Producto values (300,'Producto3',3000);

1 row created.

SQL> commit;

Commit complete.

SQL> select * from Producto;

      ID NOMBRE          PU
-----
100 Producto1          1000
300 Producto3          3000
200 Producto2          2000
```

Nota: El usuario SYS no puede utilizar esta funcionalidad en la base de datos Standby, si se intenta se conseguirá el siguiente error:

ERROR at line 1:

ORA-16397: statement redirection from Oracle Active Data Guard standby database to primary database failed

Está nueva funcionalidad también nos permite ejecutar sentencias PL/SQL desde el Standby Database y re-direccionarlas en la base de datos Primaria a través de la opción:

`SQL> ALTER SESSION ENABLE ADG_REDIRECT_PLSQL`

Ejemplo:

```
SQL> ALTER SESSION ENABLE ADG_REDIRECT_PLSQL;
Session altered.

SQL> begin
  2  execute immediate 'alter table Producto drop column pu';
  3  end;
  4  /

PL/SQL procedure successfully completed.
```

En este ejemplo queremos eliminar la columna PU (operación DDL) de la tabla Producto conectados a la base de datos Standby; el comando sigue el mismo flujo previamente explicado, es decir, primero se ejecuta en la base de datos Primaria y es replicada hacia la base de datos de Active Dataguard. Esta opción es válida siempre y cuando el código PL/SQL no utilice bind variables,

Conclusión

Esta funcionalidad ha sido muy esperada debido a que permite romper las limitantes que tenía Data Guard, dándonos la oportunidad de tener un mejor uso de las capacidades de contingencia y maximizar la inversión realizada sobre estos ambientes de naturaleza ociosa.

Publicado por:

Francisco Riccio, actualmente se desempeña como Arquitecto de Soluciones en Oracle Perú y es instructor de cursos oficiales de certificación Oracle. Es un Oracle Certified Professional en productos de Oracle Application, Base de Datos, Cloud & Virtualización.