Using Application Continuity

By Ahmed Baraka

Objectives

In this lecture, you will learn how to perform the following:

- Understand the benifits of Application Continuity
- Understand the benifits of Transaction Guard
- Describe Application Continuity Restrictions
- Create a service for Application Continuity
- Create a service for Transaction Guard

Before Application Continuity

- In case of a database (RAC or single-instance) outage:
  - The user is left in doubt: he has to check on the data changes made
  - Commit status of last transaction is unknown
  - Session state is lost
- Making changes on the applications to handle this issue is expensive and complex

What Application Continuity Can Do?

- Application Continuity masks the users from database failures
  - rebuilds the session with its state and any open transactions; and:
    - If the transaction succeeded and need not be reexecuted, the successful return status is returned to the application
    - If the transaction failed, it re-executes the transaction
    - If replay failed, error message is returned to the application
- Introduced in Oracle Database 12.1
- Works for Java applications

More about Application Continuity

- Is supported for Oracle RAC, Data Guard, Active Data Guard, and WebLogic Server
- Can be configured on the following clients:
  - JDBC Thin Oracle replay driver
  - Universal Connection Pool
  - WebLogic Server
- Better than TAF for the following reasons:
  - Transaction state is guaranteed known
  - DML operations might be replayed
  - Session state is not lost

Application Continuity Terms

Term	Description
Database Request	unit of work submitted from the application
Recoverable Error	an error that arises due to an external system failure
Commit Outcome	a transaction table is updated and a transaction is committed. transaction Guard provides a reliable commit outcome.
Mutable Object	nondeterministic function that can obtain a new value every time it is called. Examples: sequence.NextVal and SYSDATE
Session state consistency	session state after COMMIT: - Dynamic: session state cannot be fully captured (use this one) - Static: can be retrieved during a callback
In-flight Transaction	A transaction failed by an external failure with unknown status

About Transaction Guard

- Returns the outcome of the last transaction (successfully committed or not) after a recoverable error has occurred
- Applications can use its API to integrate with it: JDBC Thin, C/C++, and ODP.NET
- After outages, users can know what happened to their transactions
- Used by Application Continuity

Application Continuity Restrictions

- AC cannot be enabled or used in any request:
  - Client connection is made using the default service
  - Oracle XA applications are not supported
  - No support for Oracle deprecated classes like LOBs, ARRAY, STRUCT
- The following restrictions disable AC for part of a request:
  - If the transaction executes the **`ALTER SYSTEM`** or **`ALTER DATABASE`**
  - Active Data Guard with read/write database links to another database
- Application Continuity is not supported for logically different databases (Oracle Logical Standby and Oracle GoldenGate).

Application Continuity Restrictions (cont)

- Some actions should not be replayed (by calling disableReplay API):
  - Autonomous transactions
  - DBMS_ALERT (email or other notifications)
  - DBMS_FILE_TRANSFER (copying files)
  - DBMS_PIPE (rpc to external sources)
  - UTL_FILE (writing text files)
  - UTL_HTTP (making HTTP callouts)
  - UTL_MAIL (sending email)
  - UTL_SMTP (sending SMTP messages)
  - UTL_TCP (sending TCP messages)
  - UTL_URL (accessing URLs)

Creating Services for Application Continuity

The following service attributes must be set:

- FAILOVERTYPE=TRANSACTION
- COMMIT_OUTCOME=TRUE

Consider setting the following other parameters:

- REPLAY_INIT_TIME
- RETENTION
- NOTIFICATION=TRUE
- RLBGOAL=SERVICE_TIME
- CLBGOAL=SHORT

Creating Services for Application Continuity: Example

```
srvctl add service -db racdb -service app2
 -failovertype TRANSACTION
 -commit_outcome TRUE
 -replay_init_time 1800 -failoverretry 30 -failoverdelay 10
 -retention 86400
 -notification TRUE -rlbgoal SERVICE_TIME -clbgoal SHORT
```

Creating Services for Transaction Guard

- The following service attribute must be set:
  - **COMMIT_OUTCOME=TRUE**

```
srvctl add service -db racdb -service app3
 -commit_outcome TRUE
 -retention 86400 -failoverretry 30 -failoverdelay 10
 -notification TRUE -rlbgoal SERVICE_TIME -clbgoal SHORT
```

- The following grant must be given to the database users that retrieve the transaction status.

```
GRANT EXECUTE ON DBMS_APP_CONT TO <user name> ;
```

Summary

In this lecture, you should have learnt how to perform the following:

- Understand the benifits of Application Continuity
- Understand the benifits of Transaction Guard
- Describe Application Continuity Restrictions
- Create a service for Application Continuity
- Create a service for Transaction Guard