Maximum Application Availability with Oracle Database 12c

Oracle Development

Problems to Solve
Maximum Application Availability

Upon Database Outage Four Problems Confront Applications
1. Hang
2. Error Handling
3. Outcome of In-Flight Work
4. Resubmission of In-Flight Work
Current Situation

HA Mechanisms and Limitations

✓ Hang
  - Fast Application Notification (FAN) / Fast Connection Failover

▪ Error Handling? Not fully formalized
  - If “Connection not valid” then Application gets a New one
▪ In-Flight Work? Lost!
▪ Resubmission of In-Flight Work? N/A!

New Concepts

Recoverable Error: *is_recoverable* attribute; no need to maintain own list of
  - error codes (e.g., ora-1033, ora-1034, ora-xxx)
  - i.e., JDBC throws *SQLException*

Logical Transaction ID: to obtain COMMIT outcome

Reliable Commit Outcome: outcome upon a recoverable error.

Database Request: Unit of work submitted by the application
Database Request

Unit of Work

- PoolDataSource pds = GetPoolDataSource();
- Connection conn = getConnection(pds);
- PreparedStatement pstmt = …
  - …
  - SQL, PL/SQL, local calls, RPC
  - …
- conn.commit();
- conn.close();

Request Begins

Request Body

- Usually ends with COMMIT

Request Ends

Transaction Guard

Preserve & Retrieve COMMIT Outcome

1. Work Request
2. DB Calls
3. Errors
4. Enforce Outcome
4. Reliable Commit Outcome

- Transaction Guard allows applications to deal correctly with failures
- Without Transaction Guard, retry can cause logical corruption
- Application Continuity uses Transaction Guard
- Can be used independently via JDBC-thin, OCI/OCCI, ODP.NET

12c ORACLE Database
**Transaction Guard**

**Broad driver support in first release**

- **Commit Models**
  - Local TXN
  - Auto-commit, Commit on Success
  - Commit embedded in PL/SQL
  - DDL, DCL, Parallel DDL
  - Remote, Distributed

- **Exclusions**
  - XA
  - R/W DBLinks from Active Data Guard or Read Only

**Transaction Guard**

**Typical usage**

- **Database session outage**
  - FAN & FCF abort dead session FAST
  - Application receives an error

- If "recoverable error” then
  - Get last LTXID from dead session
  - Obtain new database session
  - Get transaction status
    
    DBMS_APP_CONT.GET_LTXID_OUTCOME with last LTXID
Transaction Guard

Configuration

- **On Service**
  - COMMIT_OUTCOME
    - Values – TRUE and FALSE
    - Default – FALSE
    - Applies to new sessions
  - RETENTION_TIMEOUT
    - Units – seconds
    - Default – 24 hours (86400)
    - Maximum value – 30 days (2592000)

---

Application Continuity

End to End Solution: Masks Outages When Successful

1. Work Request
2. DB Calls
3. Errors
4. Replay
5. Response

- Replays in-flight work on recoverable errors
- Masks many hardware, software, network, storage errors and outages when successful
- Improves end user experience
## Phases in Application Continuity

<table>
<thead>
<tr>
<th>1-Capture</th>
<th>2-Reconnect</th>
<th>3-Replay</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Keeps track of database requests</td>
<td>• Checks request has replay enabled</td>
<td>• Replays held calls</td>
</tr>
<tr>
<td>• Decides what can / cannot be replayed</td>
<td>• Handles timeouts</td>
<td>• Verifies that user visible results match original</td>
</tr>
<tr>
<td>• Holds original calls with binds and validation</td>
<td>• Creates a new connection</td>
<td>• Continues request if replay is successful</td>
</tr>
<tr>
<td>• Checks request has replay enabled</td>
<td>• Validates target database</td>
<td></td>
</tr>
<tr>
<td>• Verifies that user visible results match original</td>
<td>• Uses Transaction Guard to enforce last commit</td>
<td></td>
</tr>
<tr>
<td>• Continues request if replay is successful</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Exclusions

**Replay Disabled**

<table>
<thead>
<tr>
<th>Global</th>
<th>Request</th>
<th>Target Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Do not use default database service</td>
<td>• Restricted calls</td>
<td>• Does not support:</td>
</tr>
<tr>
<td>• Excludes XA</td>
<td>- Alter System</td>
<td>- Logical Standby</td>
</tr>
<tr>
<td>• Excludes Java deprecated concrete classes</td>
<td>- Alter Database</td>
<td>- Golden Gate</td>
</tr>
<tr>
<td></td>
<td>- Not supported for Active Data Guard with read/write database links</td>
<td></td>
</tr>
</tbody>
</table>
**AC – Application Assessment**

<table>
<thead>
<tr>
<th>Decide</th>
<th>What to do</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Boundaries</td>
<td>Mark request boundaries, if not using Oracle Pools</td>
</tr>
<tr>
<td>Side Effects</td>
<td>Use disable API if a request has a call that should not be replayed</td>
</tr>
<tr>
<td>Mutable Functions</td>
<td>Grant keeping mutable values</td>
</tr>
<tr>
<td>Callbacks</td>
<td>Register a callback for apps that change state outside requests.</td>
</tr>
<tr>
<td></td>
<td>For WebLogic and UCP labels – do nothing.</td>
</tr>
<tr>
<td>JDBC Concrete Classes</td>
<td>Replace deprecated concrete classes with Java interfaces</td>
</tr>
</tbody>
</table>

**Disabling Replay**

- **Use** `disableReplay` **API for requests that should not be replayed.**
- **Make a conscious decision to replay external actions**
  - e.g. Autonomous Transactions
  - UTL_HTTP
  - UTL_URL
  - UTL_FILE
  - UTL_FILE_TRANSFER
  - UTL_SMPT
  - UTL_TCP
  - UTL_MAIL
  - DBMS_PIPE
  - DBMS_ALERT
**Configure JDBC Replay Data Source**

- Configure the new Replay Data Source in the property file (or in the thin JDBC app)

**Use new 12.1**

```
replay datasource=oracle.jdbc.replay.OracleDataSourceImpl
```

---

**Server Configuration - Service**

- Set the service attributes:
  - FAILOVER_TYPE = TRANSACTION for using Application Continuity
  - Review the service attributes:
    - COMMIT_OUTCOME = TRUE for Transaction Guard
    - REPLAY_INITIATION_TIMEOUT = 300 after which replay is canceled
    - FAILOVER_RETRIES = 60 for the number of connection retries for each replay attempt
    - FAILOVER_DELAY = 3 for delay in seconds between connection retries
    - AQ_HA_NOTIFICATIONS = TRUE for FAN
Server Configuration - Mutables

GRANT [KEEP DATE TIME | KEEP SYSGUID].. [to USER]

- REVOKE [KEEP DATE TIME | KEEP SYSGUID][from USER]

- GRANT KEEP SEQUENCE.. [to USER] on [sequence object];

- REVOKE KEEP SEQUENCE [from USER] on [sequence object];

- If owned, ALTER SEQUENCE.. [sequence object] [KEEP|NOKEEP];
  CREATE SEQUENCE.. [sequence object] [KEEP|NOKEEP];