No Outages: Transparent Application Continuity

Knut Härtel, Pat McDevitt
Rain and Hail Crop Insurance

Carol Colrain, Ian Cookson, Troy Anthony
Oracle Database Development
October, 2018
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Program Agenda

1. What is Continuous Availability?
2. Transparent Application Continuity
3. Configuring for Best Results
4. Road to Application Continuity at Rain & Hail
From High Availability to Continuous Availability

**High Availability**

- Minimizes downtime
- In-flight work is lost
- Rolling maintenance at DB
- Predictable runtime performance
- Errors may be visible
- Designed for single failure
- Basic HA building blocks

**Continuous Availability**

- No downtime for users
- In-flight work is preserved
- Maintenance is hidden
- Predictable performance
- Errors visible only if unrecoverable
- Designed for multiple failures
- Builds on top of HA
How do we define Continuous Availability?

Customers have different definitions

Continuous Availability is not Absolute Availability.

Probable outages and maintenance events at the database level are masked from the application, which continues to operate with no errors and within the specified response time objectives while processing these events.

Key points:

1. Planned maintenance and likely unplanned outages are hidden from applications
2. There is neither data loss nor data inconsistency
3. Majority of work (% varies by customer) completes within recovery time SLA
4. May appear as a slightly delayed execution

Many customers are achieving Continuous Availability Today
What kinds of outages?

Planned Maintenance
Patches Repairs Upgrades Changes

Unplanned Outages

Unpredictable Response & Throughput

Site Disasters

Data Corruption

Human Errors

Which outage classes does your business need to handle?
Applications should see no errors during outages

Before Application Continuity

Database outages cause in-flight work to be lost, leaving users and applications in-doubt

- Restart applications and mid-tiers
- User frustration
- Cancelled work
- Duplicate submissions
- Errors even when planned
- Developer pains
New Concept - Request

All Oracle 12c Pools and in JDK9

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

Standardize on TAC
Applications see no errors during outages

Standardize on Transparent Application Continuity

Hides errors, timeouts, and maintenance

No application knowledge or changes to use

Rebuilds session state & in-flight transactions

Adapts as applications change: protected for the future
TAC Explained

**Normal Operation**
- Client marks requests: explicit and discovered.
- Server tracks session state, decides which calls to replay, disables side effects.
- Directed, client holds original calls, their inputs, and validation data.

**Failover Phase 1: Reconnect**
- Checks replay is enabled
- Verifies timeliness
- Creates a new connection
- Checks target database is legal for replay
- Uses Transaction Guard to guarantee commit outcome

**Failover Phase 2: Replay**
- Restores and verifies the session state
- Replays held calls, restores mutables automatically
- Ensures results, states, messages match original.
- On success, returns control to the application

New with TAC
# Using Transparent Application Continuity

<table>
<thead>
<tr>
<th>Request Boundaries</th>
<th>Discovered and Advance Continuously</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session State</td>
<td>Restored and Verified</td>
</tr>
<tr>
<td>Side Effects</td>
<td>Not replayed</td>
</tr>
<tr>
<td>SYSDATE, Sequences..</td>
<td>Automatic for SQL, Grant for PL/SQL</td>
</tr>
<tr>
<td>Coverage</td>
<td>Always know your protection level</td>
</tr>
</tbody>
</table>
Request Boundaries Advance Continuously

• Request boundaries advanced automatically when state is restorable*
• Capture re-enables, if disabled
• Smaller capture set means faster recovery
• Oracle pool and return to pool are still best practice

* Transparent Application Continuity for Java
Session State must be Correct to Replay

• Restore session states before replaying
  – `FAILOVER_RESTORE` on your service

• TAC does not replay if session states differ

• AC allows you to add complex states before replay starts
  – AC does not replay if standard states differ
Side Effects Not Replayed

**TAC – stops capture automatically until next enable point**

TAC decides if any requests should not be replayed, e.g.

- UTL_HTTP
- UTL_URL
- DBMS_FILE
- DBMS_FILE_TRANSFER
- UTL_SMTP
- UTL_TCP
- UTL_MAIL
- EXTPROC

**Side Effects Disabled**

Customized: use AC
Handles SYSDATE, SYSTIMESTAMP, Sequences ...

Keeps and Restores Automatically

Function results kept for SQL

Grant keeping results for PL/SQL

For owned sequences:

- ALTER SEQUENCE.. [sequence] [KEEP|NOKEEP]
- CREATE SEQUENCE.. [sequence] [KEEP|NOKEEP]

Grant and Revoke for other users:

- GRANT [KEEP DATE TIME | KEEP SYSGUID] [to USER]
- REVOKE [KEEP DATE TIME | KEEP SYSGUID] [from USER]
- GRANT KEEP SEQUENCE on [sequence] [to USER]
- REVOKE KEEP SEQUENCE on [sequence] [from USER]
Always Know Your Protection Level

- AWR, system, session, service stats
  - Requests completed per second
  - User calls in request
  - Protected user calls

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Total</th>
<th>per Second</th>
<th>per Trans</th>
</tr>
</thead>
<tbody>
<tr>
<td>cumulative requests</td>
<td>177,406</td>
<td>49.2</td>
<td>5.0</td>
</tr>
<tr>
<td>cumulative user calls in request</td>
<td>493,329</td>
<td>136.8</td>
<td>13.8</td>
</tr>
<tr>
<td>cumulative user calls protected</td>
<td>493,329</td>
<td>136.8</td>
<td>13.8</td>
</tr>
</tbody>
</table>
## Application Continuity Summary

<table>
<thead>
<tr>
<th>Outage Type</th>
<th>Status</th>
<th>Message</th>
</tr>
</thead>
</table>
| Coverage checks   | TotalRequest = 398  
PASS = 389  
WARNING = 0  
FAIL = 9          | [PASS] Trace file name = WEB ora 124333.trc Row number = 10909  
SERVICE NAME = (WEB_SSL SERVICE) MODULE NAME = (JDBC Thin Client) ACTION NAME = POST  
CLIENT ID = null  
Coverage(%) = 100 ProtectedCalls = 1 UnProtectedCalls = 0 |
| PASS              | [PASS] Trace file name = WEB ora 19757.trc Row number = 36978  
SERVICE NAME = (WEB_SSL SERVICE) MODULE NAME = (JDBC Thin Client) ACTION NAME = CARD  
CLIENT ID = null  
Coverage(%) = 100 ProtectedCalls = 24 UnProtectedCalls = 0 |
| FAIL              | [FAIL] Trace file name = WEB ora 19757.trc Row number = 481193  
SERVICE NAME = (WEB_SSL SERVICE) MODULE NAME = (JDBC Thin Client) ACTION NAME = null  
CLIENT ID = null  
Coverage(%) = 20 ProtectedCalls = 1 UnProtectedCalls = 4 |
| FAIL              | [FAIL] Trace file name = WEB ora 19757.trc Row number = 14203  
SERVICE NAME = (WEB_SSL SERVICE) MODULE NAME = (JDBC Thin Client) ACTION NAME = null  
CLIENT ID = null  
Coverage(%) = 33 ProtectedCalls = 2 UnProtectedCalls = 1 |
Configuration at Database

Service Attributes

- `FAILOVER_TYPE = AUTO`
- `FAILOVER_RESTORE = AUTO`
- `COMMIT_OUTCOME = TRUE`
- `AQ_HA_NOTIFICATIONS=True` for FAN OCI
- `REPLAY_INITIATION_TIMEOUT = 300` - seconds before replay is canceled
Are your building blocks in place?
Services for Location Transparency

Services provide a “dial in number” for your application

Regardless of location, application keeps the name

Moving, reshaping, prioritizing controls how a service is offered

Batch and OLTP separated

DB and PDB names for admin only

Node 1

Node 2

RAC instance

RAC instance

Listener

OLTP service

Batch service
Connections Appear Continuous

Standard for All Drivers from 12.2

```
alias = (DESCRIPTION = 
    (CONNECT_TIMEOUT=90)  (RETRY_COUNT=20)(RETRY_DELAY=3)  
    (TRANSPORT_CONNECT_TIMEOUT=3)  
    (ADDRESS_LIST =  
        (LOAD_BALANCE=on)  
        ( ADDRESS = (PROTOCOL = TCP)(HOST=primary-scan)(PORT=1521))))  
    (ADDRESS_LIST =  
        (LOAD_BALANCE=on)  
        ( ADDRESS = (PROTOCOL = TCP)(HOST=secondary-scan)(PORT=1521))))  
    (CONNECT_DATA=(SERVICE_NAME = gold-cloud)))
```

Automatic Retries

Configure in One Place

Single description

No reliance on DNS

ALWAYS use a SERVICE that is NOT DB/PDB name

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |
FAN for INSTANT Interrupt

The dead thing cannot tell you it is dead

All Oracle uses FAN

<table>
<thead>
<tr>
<th>JDBC Universal Connection Pool</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCI/OCCI driver</td>
</tr>
<tr>
<td>ODP.NET Unmanaged Provider (OCI)</td>
</tr>
<tr>
<td>ODP.NET Managed Provider (C#)</td>
</tr>
<tr>
<td>OCI Session Pool</td>
</tr>
<tr>
<td>WebLogic Active GridLink</td>
</tr>
<tr>
<td>Tuxedo</td>
</tr>
<tr>
<td>JDBC Thin Driver (new 12.2)</td>
</tr>
<tr>
<td>CMAN and Listeners</td>
</tr>
</tbody>
</table>

Auto-Configured in 12c

DESCRIPTION =

(CONNECT_TIMEOUT=90)
(RETRY_COUNT=20) (RETRY_DELAY=3)
 TRANSPORT_CONNECT_TIMEOUT=3)

(ADDRESS_LIST =
 (LOAD_BALANCE=on)
 ( ADDRESS = (PROTOCOL = TCP)
 (HOST=primary-scan) (PORT=1521)))

(ADDRESS_LIST =
 (LOAD_BALANCE=on)
 ( ADDRESS = (PROTOCOL = TCP)
 (HOST=second-scan)(PORT=1521)))

(CONNECT_DATA=(SERVICE_NAME=gold))

ONS Node Set 1

ONS Node Set 2
Recover in a Timely Manner

**TUNABLES**

- **MISSCOUNT**
  - Hardware & adjustable

- **FAST_START_MTTR_TARGET**

- **Flex ASM**

- **SCSI Timeout**

- **Service Isolation (from 18c)**

- **PING_TIMEOUT**

- **FSFO Observer**

**Node Panic** – non-master

**Node Panic** – master

**Public Network**

**Instance Join**

**Dynamic Remastering**

**Private Interconnect**
Road to Application Continuity

Knut Härtel, DBA
Pat McDevitt, AVP IT
October, 2018
Agenda

• Who we are
• Business objectives
• Implementation steps
• Application Continuity at Work
• Tips and Tricks – Not to Miss
• Conclusion
Who we are

• Subsidiary of Chubb and support the agribusiness line of business

• Chubb is the world’s largest publicly traded P&C insurance company and the largest commercial insurer in the U.S.

• We provide commercial and personal property and casualty insurance, personal accident and supplemental health insurance, reinsurance and life insurance to a diverse group of clients.

• operations in 54 countries and territories
Our Oracle Journey

• Oracle Customer for more than 20 years

• Started with Oracle OPS on OpenVMS - 1995

• Migration to 9i RAC with Veritas-Cluster on Solaris - 2008
  • Introduction of select failover with JDBC OCI and TAF, for unplanned outages

• Migration to 11g RAC on OEL UEK Intel - 2013
  • Introduced OID for TNS lookup

• Database Upgrade to 12.1 - 2017

• Java is our main development language for Oracle since 1999
Business objectives

• Upgrade the clients to avoid extended license fees
• Increase return on investment for software license – App Continuity is incl. with RAC
• Support data types that our Applications need
• Recover transactions, which are our business
# Client Decision Matrix

<table>
<thead>
<tr>
<th>Client/HA Features</th>
<th>JDBC OCI 12.1</th>
<th>JDBC THIN 12.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool</td>
<td>UCP</td>
<td>UCP</td>
</tr>
<tr>
<td></td>
<td>3rd party pool</td>
<td>3rd party pool</td>
</tr>
<tr>
<td>TAF</td>
<td>No LOB Types</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>No LOB Types</td>
<td>x</td>
</tr>
<tr>
<td>FCF/FAN</td>
<td>No FCF</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>No FCF</td>
<td>No FCF</td>
</tr>
<tr>
<td>Transaction Idempotence</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Application Continuity</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

> JDBC-THIN with UCP and AC gives best return on investment
Easy Implementation Steps for Application Continuity - Application

1. Use Oracle Pool

2. Basic principle must apply
   - When a request is completed, return the connection

3. Check and remove deprecated JAVA classes (prior 18c)
   - deprecated oracle.sql concrete classes like BLOB, CLOB, BFILE, OPAQUE, ARRAY, STRUCT or ORADATA (we only had a couple)
Easy Implementation Steps for Application Continuity - Application Server

• Configure FAN and FCF with ONS
  • Starting from UCP 12.2 enabled by default

• depending on your Application needs, set autoCommit to false through connection properties
  • `connectionProperties=" autoCommit=false "`

• Use Recommended TNS Format for High Availability to Autoconfigure FAN
  • `(CONNECT_TIMEOUT=120) (RETRY_COUNT=20) (RETRY_DELAY=3) (TRANSPORT_CONNECT_TIMEOUT=3)`
  • When Data Guard in use, add `address_list` for Data Guard
  • `tnsnames.ora` with JDBC-Thin, use `oracle.net.tns_admin`

• Pool timeout settings
  • `connectionWaitTimeout` needs to be higher than `Connect_Timeout`!
Easy Implementation Steps for Application Continuity - RAC-Database

- Fast Application Notification
  - Out of the box configured starting from Version 12.1, when using recommended TNS SCAN-Listener
  - Needed on RAC for providing continuous Connectivity

- Standardize on database services that support Application Continuity

- Set FAST_START_MTTR_TARGET to meet Recovery Time Objective

- Grants for mutable objects
  - Sysguid, Systimestamp/Sysdate, Sequences
Easy Implementation Steps for Application Continuity - Configuration

Initial Configuration
- Database 12c Version 12.1.0.2
- Client 11g Version 11.2.0.4
  - Instant Client JDBC OCI
- Tomcat
  - dbcp Pool
  - Default autocommit true
- JAVA 1.8.0 102

Final Configuration
- Database 12c Version 12.1.0.2
- Client 12c Version 12.2.0.1
  - JDBC THIN
- Tomcat
  - UCP Pool (FAN,FCF)
  - Autocommit false
- JAVA 1.8.0 102
Easy Implementation Steps for Application Continuity - Architecture

- 400 Application Servers per site
- Database schema per crop year
- 18 Database services

Application Servers

SCAN

- OLTP-SVC
- OLTP-SVC
- DW-SVC

RAC Primary

Application Servers

Listener

- Standby

Data Guard Standby

ONS
AC at Work - Unplanned outage

Unplanned outage of Node2.

Chart is based on hourly AWR Snapshot

Unplanned outage of Node2.
AC at Work - Unplanned Outage

Unplanned outage of Node2

Chart is based on hourly AWR Snapshot
AC at Work - Planned maintenance

Chart is based on hourly AWR Snapshot

AWR - User Transactions per Sec across all DB Services

Draining
Begin
Maintenance
End

Rain and Hail
A Chubb Company

October 24, 2018
Tips and Tricks

• Configure timeouts correctly

• Recovery Point Objective of SLA

• Timeouts and settings that need to be verified
  • Cluster (CSS_MISSCOUNT = 30)
  • Database (FAST_START_MTTR_TARGET = 30)
  • TNS Connect Timeout = 120
  • UCP Connection Wait Timeout = 130
  • Service Replay Initiation Timeout = 300

• Fast Sync for Data Guard (MAA / 12c)
  • Add FAST_START_FAILOVER of Data Guard and configure Observer

• UCP Connection Wait Timeout for getting a connection needs to be higher as the TNS Connect Timeout, otherwise the pool can cancel reconnect requests too early during reconnect
Tips and Tricks I

UCP ConnectionWaitTimeout

TNS Connect Timeout

SCAN

DB-Service

CSS_Misscount

Node1

RAC Primary
FAST_START_MTTR_TARGET

Node2

FSFO

Fast Sync

Standby

DB-Service

SCAN

Data Guard
FAST_START_MTTR_TARGET
Tips and Tricks II

- Set v$session.Program through Pool Settings
  - This in turn, sets the v$session.MODULE which provides granular information in ASH and other v$ views

- Add own versioning file, that includes applied patches, into the jar
  - The versioning file helps to identify which version is in use

- Starting with 12c JDBC Thin, the date includes as well the time
  - oracle.jdbc.DateZeroTime=true to keep previous behavior

- Application Continuity Memory Management
  - Apply patch for Bug 28538380
Tips and Tricks II

• Batch Application
  • Apps that did 64k individual inserts as one transaction!
  • A maximum of Round trips between Client and Database per Request of 65536
  • Moved to addBatch JAVA function and it used just one round trip

• Multiple transactions in same request
  • 12.1 is for OLTP, replay is disabled after the 1st commit
  • Multiple commits supported with AC static mode in 12.2 or with TAC in 18c (preferred)

• Use ‘order by’ on queries (same as TAF) as a good practice
• AC checklist on oracle.com/goto/ac
Our Oracle Journey Continues

• Achievements
  • Client upgrade to 12.2
  • Introduction of Application Continuity for unplanned Outages
  • Ability to perform planned maintenance 24x7
  • Centralized tnsnames.ora

• What’s Next
  • Implement Active Data Guard with FAST SYNC
  • Upgrade to 18c database
Conclusions

• Easy and straightforward Implementation
• Test often, test regularly, test thoroughly

“Being an Insurance Company, Application Continuity provides us the coverage that we need for our business.”
Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Integrated Cloud
Applications & Platform Services