



Oracle OpenWorld 2019

SAN FRANCISCO





Zero Downtime for Java Applications: What's New, Use cases and Solutions



Nirmala Sundarappa

Principal Product Manager
Oracle JDBC and UCP

Kuassi Mensah

Director, Product Management
Oracle JDBC and UCP

Jean De Lavarene

Director, Product Development
Oracle JDBC and UCP


Safe Harbor

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, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at <http://www.oracle.com/investor>. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.

Agenda



- 
- 1 Announcement
 - 2 Building Blocks of Continuous Availability
 - 3 Transparent Application Continuity (TAC) – Planned/Unplanned
 - 4 Transparent Application Continuity with ATP-D
 - 5 Questions

19.3 JDBC drivers & some companion jars on Central Maven

<https://repo1.maven.org/maven2/com/oracle/ojdbc/>

`<groupId>com.oracle.ojdbc</groupId>`

`<artifactId>ojdbc8</artifactId>`

`<version>19.3.0.0</version>`

Older releases will be available later



Building Blocks of continuous availability

- Application side
 - Universal Connection Pool (UCP)
 - Recommended connection URL
 - Fast Application Notification (FAN)
 - Request Boundaries
 - Transparent Application Continuity (TAC)
- Database Server side
 - MAA Architecture: RAC / ADG / DG
 - Use Database Services

UCP with other Java-based Application Servers

```
<Context docBase="ATPWebApp" path="/ATPWebApp"
  reloadable="true" source="org.eclipse.jst.jee.server:samplejdbcpage">

  <Resource name="tomcat/UCP_atp" auth="Container"
    factory="oracle.ucp.jdbc.PoolDataSourceImpl"
    type="oracle.ucp.jdbc.PoolDataSource"
    description="UCP Pool in Tomcat"
    connectionFactoryClassName="oracle.jdbc.pool.OracleDataSource"
    minPoolSize="5"
    maxPoolSize="50"
    initialPoolSize="15"
    user="hr"
    password="hr"
    url="jdbc:oracle:thin:@databasename_medium?TNS_ADMIN=/Users/test/lib"
  />

</Context>
```

- IBM WebSphere
- IBM Liberty
- Apache Tomcat
- NEC WebOTX
- Red Hat WildFly (JBoss)
- Hibernate
- Spring
- custom

Recommended Connection String

All drivers 12.2

Automatic Retries

Configure Once
in LDAP or
tnsnames.ora

```
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)))
```

No reliance on DNS

ALWAYS use a service that is NOT DB/PDB name

Concept: Fast Application Notification (FAN)

- The FAN allows applications to be notified instantaneously of Oracle Database events
- Supported FAN events are mentioned below
 - **Node Events**: Node Down, Public Network Down
 - **Instance Events**: Instance Down, Instance Up
 - **Service Events**: Service member Down, Service member Up, Service Down, Service Up
 - **Database Events**: Database Down, Database Up
- The Universal Connection Pool (UCP) and JDBC driver support FAN (Without any application changes)

Sample FAN event

- SERVICE DOWN
- VERSION=1.0 **event_type=SERVICE**
service=testy_pdb_srv.us.oracle.com database=testy
db_domain=us.oracle.com host=rachost_723 **status=down**
reason=USER timestamp=2019-01-21 19:25:52 timezone=-08:00

Concept: Request Boundaries

UCP 12c+, Java Standard (JDK9+), Transparent Application Continuity (TAC)

```
PoolDataSource pds = GetPoolDataSource();  
Connection conn = pds.getConnection();  
PreparedStatement pstmt = ...
```

```
...  
SQL, PL/SQL, local calls, RPC  
...
```

```
conn.commit();  
conn.close();
```

**Begin
Request**

Request Body
often ends with
COMMIT

**End
Request**

Application Continuity Explained

Normal Operation

Client marks requests: explicit and **implicit**.

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 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 application

Evolution of Application Continuity (AC)

18c

Transparent Application Continuity (TAC) in the JDBC driver

- Transparent in most cases
- Detects & reports implicit request boundaries to the server
 - `oracle.jdbc.enableImplicitRequest` to turn off
- Handles state signature at runtime and replay
- Support for concrete classes

19c

- Implicitly starts requests using Java standard APIs on fresh connections
- **TAC is ON by default on ATP-D**
Use TP and TPURGENT for TAC
- Full session parameters restored, use `FAILOVER_RESTORE` on server side

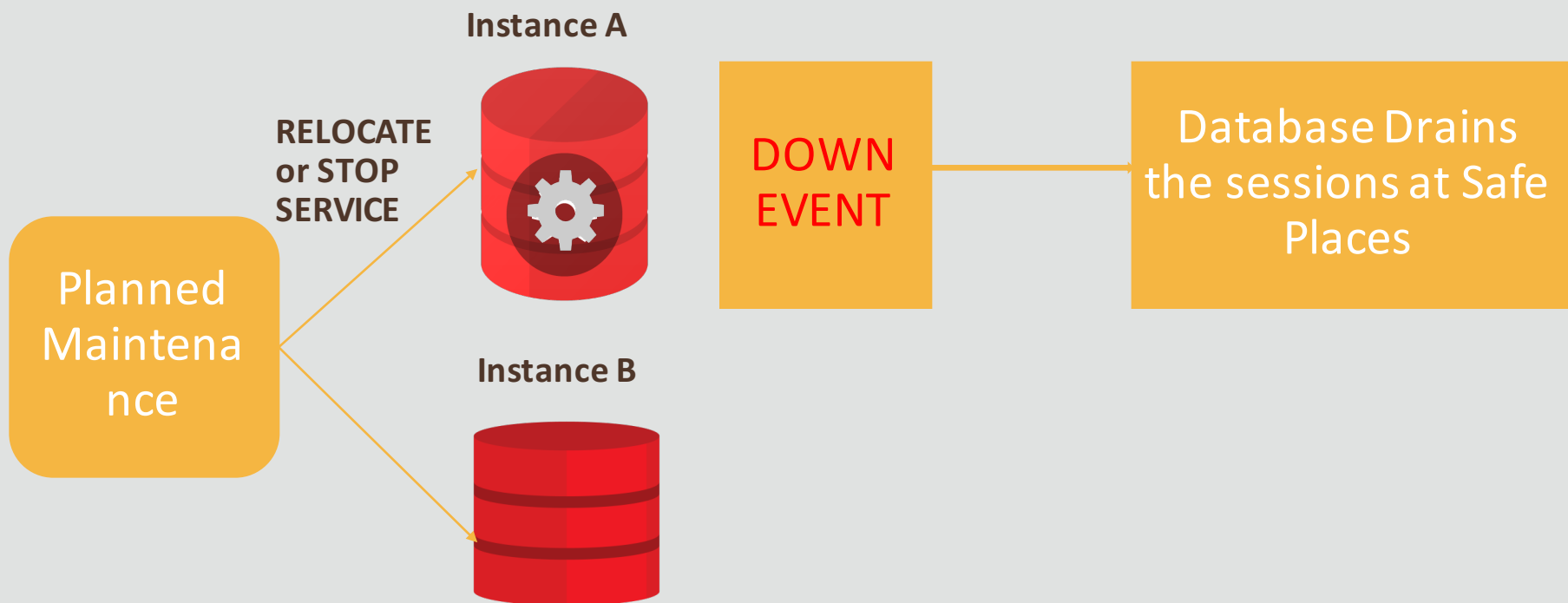
20c

- MODP (Modifiable Parameters) state coverage for both TAC and AC
- TAC template and overflow support
- Server has "planned failover"

AC support for concrete classes

- The 18.3 JDBC driver supports the following concrete classes with Application Continuity
 - `oracle.sql.CLOB`,
 - `oracle.sql.NCLOB`
 - `oracle.sql.BLOB`
 - `oracle.sql.BFILE`
 - `oracle.sql.STRUCT`
 - `oracle.sql.REF`
 - `oracle.sql.ARRAY`
- NOT SUPPORTED datatypes
 - `oracle.sql.OPAQUE` and `oracle.sql.ANYDATA`

Planned maintenance - Draining



Planned Node Maintenance with JDBC

DB 12.2 JDBC driver (`ojdbc8.jar`)

- Driver closes the database sessions at a “Safe Place” (See connection validation options)
- Application must use one of the **safe draining methods** (next slide)
- Requires **`simplefan.jar`** and **`ons.jar`** in the classpath

Tip: Enable Connection Tests for Application Servers

Application Server	Test Name	Connection Test to DB
Oracle WebLogic – Generic and Multi data sources	TestConnectionsOnReserve TestConnectionsOnCreate	isUsable() SQL – SELECT 1 FROM DUAL
Oracle WebLogic Active GridLink	Embedded	isUsable()
IBM WebSphere	PreTest Connections	SQL – SELECT 1 FROM DUAL
RedHat WildFly (JBoss)	Check-valid-connection-sql	SQL – SELECT COUNT(*) FROM DUAL
Apache Tomcat	TestOnBorrow TestOnRelease	SQL – SELECT 1 FROM DUAL

Planned Node Maintenance

Java Requirements (Add these in the classpath)

- Option A: DB 12.2 + `ucp.jar`
- Option B: DB 12.2 + `ojdbc8.jar`
- Option C: DB 12.2 + `ons.jar` + `simplefan.jar` if not using UCP

Transparency for all Outages



Planned Maintenance

Patches
Repairs
Upgrades
Changes



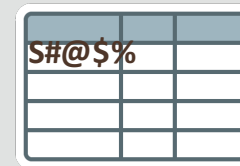
Unplanned Outages



Unpredictable Response & Throughput



Site Disasters



Data Corruption



Human Errors

Solutions for handling Unplanned Outage

- Use UCP with replay datasource
 - `oracle.jdbc.replay.OracleDataSourceImpl`
- Use APIs to check the AC statistics
 - `OracleDataSource.getReplayStatistics(StatisticsReportType)`
 - `OracleDataSource.getReplayStatistics()`
- For server side settings, refer to the whitepaper or AC documentation

Best Practices for handling Unplanned Outage

- Return connections to the pool
 - Check out and check in when the DB work is complete
 - Allows AC to embed request boundaries at safe places
- Enable JDBC Statement caching
 - `ods.setImplicitCachingEnabled(true);`
 - `(OracleConnection) conn).setStatementCacheSize(10);`
 - Disable the statement caching provided by the app servers (WLS, Tomcat etc.,)
- Align Application and Server Timeouts
 - `oracle.jdbc.ReadTimeout` and `HTTP_REQUEST_TIMEOUT` should be higher than detection and recovery timers on the server side

High Availability with ATP-D

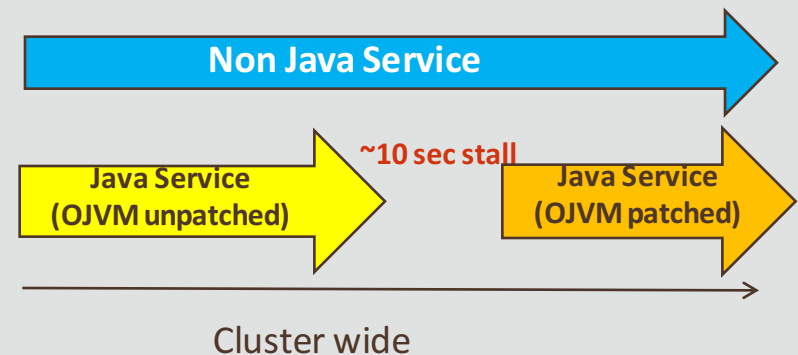
- Transparent Application Continuity is enabled by default on ATP-D
 - Use `tpurgent` or `tpurgent_tls` services
- Use the replay datasource (`oracle.jdbc.replay.OracleDataSourceImpl`)
- Set the ONS configuration on the client side
 - `onsConfiguration="nodes=<ATPDhost>:6200\nwalletfile=/net/host/path/onswallet"`
- Make sure to have *ons.jar* and *simplefan.jar* on the classpath
- Increase the connection timeouts and *retry_count* and *retry_delay*
- RAC Rolling restart will handle everything on the server side

Transparent Application Continuity with ATP-D (DEMO)

OJVM Rolling Upgrade- Planned Maintenance -overview

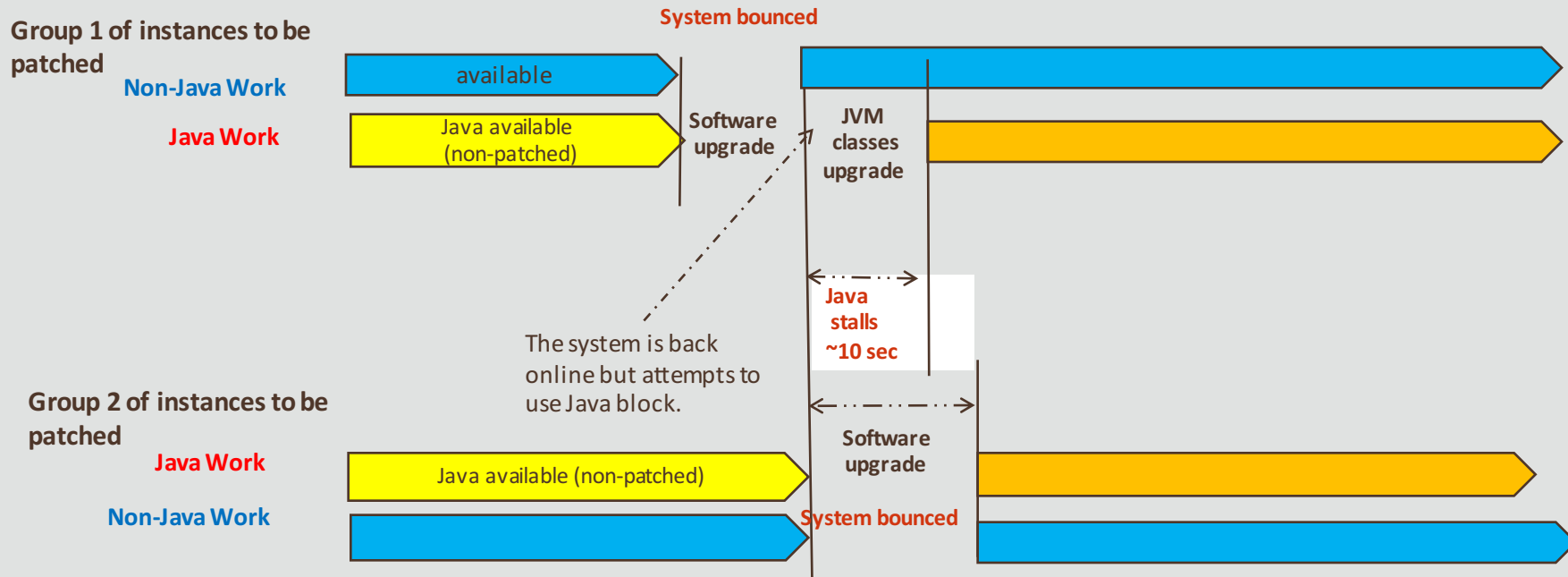
DB 18.4: OJVM Patching will be RAC Rolling

- No functional change
- Single PSU for both DB and OJVM
- Non-Java service available all time (at least one instance)
- Java service available all time, except ~10 sec brown-out
- No error reported during brown-out
- CDB/PDBs supported (may incur longer brown-out)
- Looking into back-porting to 11.2 and 12.1 (plan not finalized)
- White paper + MOS Note



OJVM Rolling Upgrade- Planned Maintenance - Workflow

- 2 or more groups of instances targeted for upgrade
 - JVM classes upgrade -> once per database or CDB
 - Software upgrade -> each instance



Thank You

Oracle JDBC and UCP

Product Management and Product Development
www.oracle.com/jdbc

