
ORACLE®

Zero Brownout Mantra for Applications using Real Application Clusters

Carol Colrain, Kevin Neel, Troy Anthony

Consulting Members of Technical Staff, ST Development

ORACLE DATABASE

ORACLE

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. Problems to Solve
2. Fast Application Notification
3. Continuous Connections
4. Hiding Planned Maintenance
5. Hiding Unplanned Outages
6. Success Stories

1 What problems confront applications at database outages?

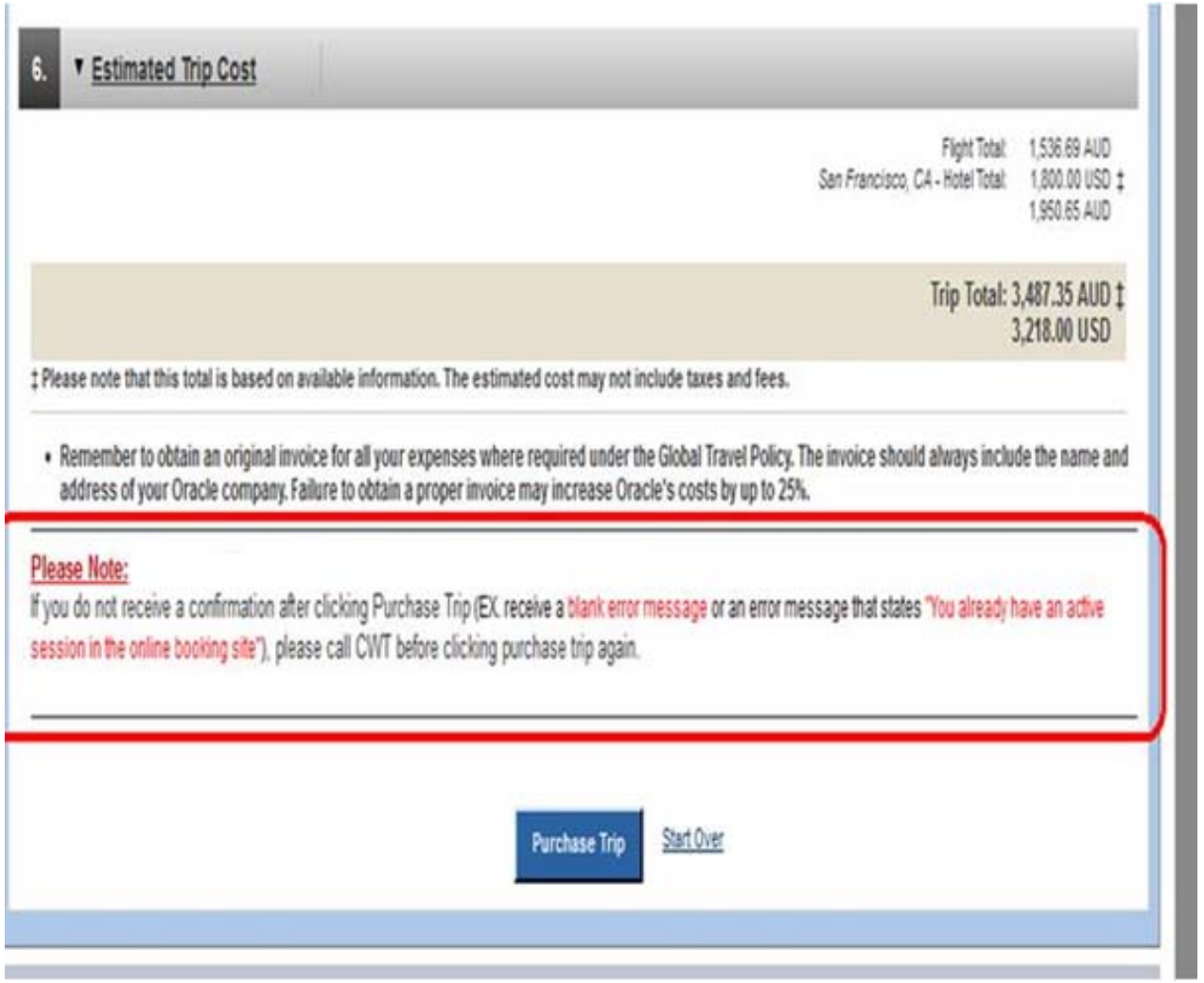
In-Flight Work

Sorry. Internal Server Error - 500 Error
We are currently experiencing an issue with our servers on coolcar.com. Please come back later.

Pre-12c Situation

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



How do we reach all applications?

- Move work to different instance/database with no errors reported to applications at planned maintenance
- Hide unplanned database outages from the applications
- Take adoption out of the developers hands to configuration/operation only
- Work with current drivers and older database, whenever possible

2 Outage Detection

The dead thing cannot tell you that it's dead

Applications Waste Time

- Hanging on TCP/IP timeouts
- Connecting when services are down
- Not connecting when services resume
- Receiving errors during planned maintenance
- Processing partial results when server is down
- Attempting work at slow, hung, or dead nodes

Fast Application Notification

Proven since 10g

- **Down** – received in low ms to invoke failover
- **Planned Down** – drains sessions for planned maintenance with no user interruption whatsoever
- **Up** – Re-allocates sessions when services resume
- **Load %** - Advice to balance sessions for RAC locally and GDS globally
- **Affinity** - Advice when to keep conversation locality

12c: Auto-Configuration + Global Data Services

12c FAN: Standardized, Auto-Configured

Client	10g	11g	12c
JDBC Implicit Connection Cache	ONS	ONS	desupport
JDBC Universal Connection Pool		ONS	**ONS**
OCI/OCCI driver	AQ	AQ	**ONS**
ODP.NET Unmanaged Provider (OCI)	AQ	AQ	**ONS**
ODP.NET Managed Provider (C#)		ONS	**ONS**
OCI Session Pool	AQ	AQ	**ONS**
WebLogic Active GridLink		ONS	**ONS**
Tuxedo		ONS	**ONS**
Listener	ONS	ONS	**ONS**

12c JDBC FAN Auto-Configures

- **12c JDBC clients and 12c Oracle database**
  - Check ons.jar is included in the class path
  - To enable FAN set the pool property
    - **fastConnectionFailoverEnabled=true**
- **Before 12c - JDBC clients or database**
  - also set the pool property for remote ons
    - **ONSConfiguration=nodes=mysun05:6200,mysun06:6200, mysun07:6200,mysun08:6200**

12c OCI FAN Auto-Configures

- **12c OCI clients and 12c Oracle database**

  Use srvctl to configure the service for AQ HA Notification:

  srvctl modify service -db EM -service GOLD **-notification TRUE**

  For the client, enable in oraaccess.xml

```
<oraaccess>
  <default_parameters>
    <events>true</events>
  </default_parameters>
</oraaccess>
```

- **Before 12c OCI clients or database**

  - Enable OCI_EVENTS at environment creation OCIEnvCreate(..)
  - Link the app with the client thread o/s library.

12c ODP.Net FAN Auto-Configures

- **12c ODP.Net clients and 12c Oracle database**

  Use srvctl to configure the service for AQ HA Notification:

  srvctl modify service -db EM -service GOLD **-notification TRUE**

  To enable FAN, in the connection string -

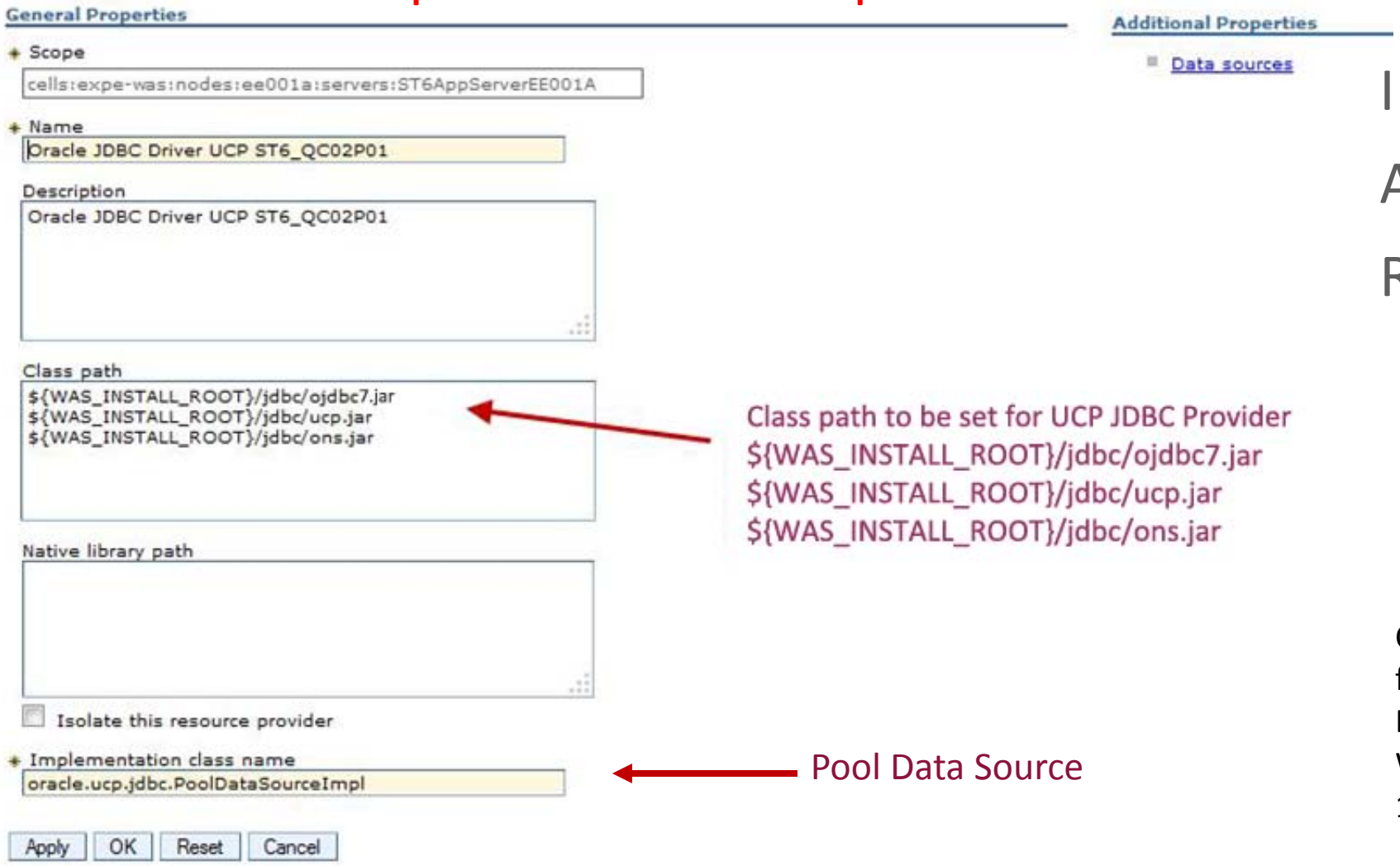
  – **"user id=oracle; password=oracle; data source=HA; pooling=true; HA events=true;“**

  To enable Runtime Load Balancing, also in the connection string -

  – **"user id=oracle; password=oracle; data source=HA; pooling=true; HA events=true; load balancing=true;“**

FAN with other Java Application Servers

Use UCP – a simple DataSource replacement



Class path to be set for UCP JDBC Provider
${WAS_INSTALL_ROOT}/jdbc/ojdbc7.jar
${WAS_INSTALL_ROOT}/jdbc/ucp.jar
${WAS_INSTALL_ROOT}/jdbc/ons.jar

Pool Data Source

IBM WebSphere

Apache Tomcat

Red Hat Jboss

CON7757 - Design Applications for Planned and Unplanned Database Downtime - Wednesday, Oct 1, 11:30 AM - 12:15 PM - Moscone South - 308

Monitor FAN

- **Create a FAN callout in ..$GRID_HOME/racg/userco**
- **Download ONS subscriber (ONCCTL) from OTN RAC page**

oncctl

..

VERSION=1.0 **event_type=SERVICEMEMBER** service=orcl_swing_pdb2 instance=orcl1 database=orcl db_domain= host=sun01 **status=down reason=USER** timestamp=2014-07-30 12:02:51 timezone=-07:00

VERSION=1.0 event_type=**SERVICEMEMBER** service=orcl_swing_pdb10 instance=orcl1 database=orcl db_domain= host=sun01 **status=down reason=USER** timestamp=2014-07-30 12:02:52 timezone=-07:00

VERSION=1.0 event_type=SERVICE service=orcl_swing_pdb10 database=orcl db_domain= host=sun01 status=down **reason=USER**

Continuous Connections

Applications should see no errors while services relocate.

Connections Appear Continuous

while a service is temporarily unavailable to receive work

New

Safe for logon storms

Retry while service is unavailable

```
alias =(DESCRIPTION =
    (CONNECT_TIMEOUT=90) (RETRY_COUNT=30)(RETRY_DELAY=10)
        (TRANSPORT_CONNECT_TIMEOUT=10)
    (ADDRESS_LIST =
    (LOAD_BALANCE=on)
    (ADDRESS = (PROTOCOL = TCP)(HOST=primary-scan)(PORT=1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST=secondary-scan)(PORT=1521)))
    (CONNECT_DATA=(SERVICE_NAME = gold-cloud)))
```

OCI only

Transparent Planned Maintenance

Applications should see no errors during maintenance.

Transparent Planned Maintenance

- Pools drain work away from instances targeted for maintenance initiated by FAN
  - Supports well behaved applications using Oracle pools
    - WebLogic Active GridLink, UCP, ODP.NET unmanaged and managed, OCI Session Pool, PHP
    - 3rd party application servers using UCP DataSource: IBM Websphere, Apache Tomcat,..
- Failover at transactional disconnect
    - applications adapted for TAF SELECT with OCI or ODP.Net unmanaged provider
    - applications with own/custom failover

DBA steps - Drain Work at Safe Places

Repeat for each service allowing time to drain

- **Stop service (no –force)**

```
SRVCTL stop service –db .. -instance .. -service ..
```

- or **Relocate service (no –force)**

```
SRVCTL relocate service –db .. -service .. –oldinst .. –newinst
SRVCTL relocate service –db .. -service .. –currentnode.. –targetnode
```

- **Wait to allow sessions to drain,** e.g. 10-30 minutes
- **Stop the sessions transactional for remaining sessions**

```
exec dbms_service.disconnect_session( '… your service .',
 DBMS_SERVICE.POST_TRANSACTION);
```

- **Now stop the instance immediate; option to disable**

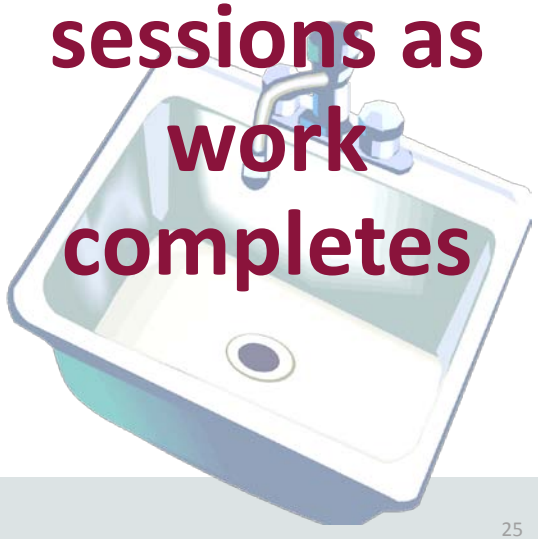
CON8176 - Rapid Home Provisioning – Thu 9:30 AM - 10:15 AM Moscone North - 131

How it works

Applications using …	Oracle pools or drivers – WebLogic Active GridLink, UCP, ODP.NET managed/unmanaged, OCI, Tuxedo 3rd party App Servers using UCP: IBM WebSphere, Apache Tomcat, RedHat JBoss
DBA Step	**srvctl [relocate|stop] service (no –force)**
Sessions Drain	Immediately New work is redirected by listeners Idle sessions are released Active sessions are released when returned to pools

FAN Planned

Pools drain sessions as work completes



Planned Maintenance at NEC

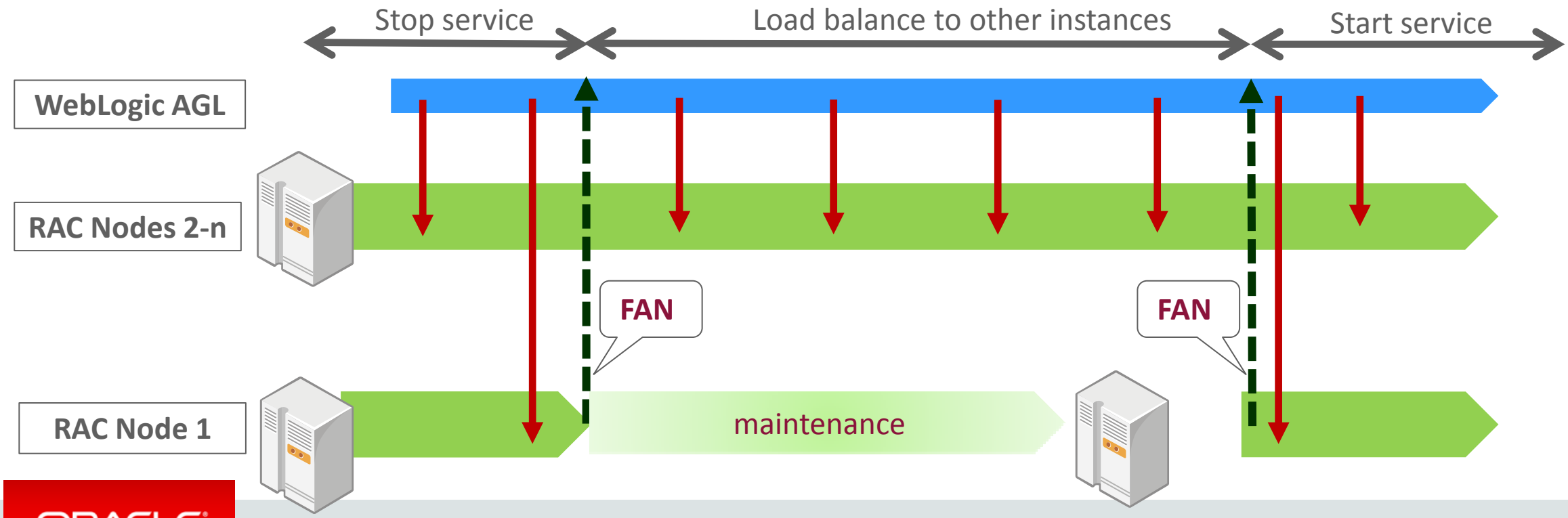
WebLogic Active GridLink and Real Application Clusters

Empowered by Innovation

NEC

1. srvctl stop services at one instance & drain **(e.g. 5-7s)**
2. Instance shutdown
3. Apply patch or change parameter or other maintenance
4. Restart instance & service

No errors, application continues



ORACLE

Planned Maintenance at NEC

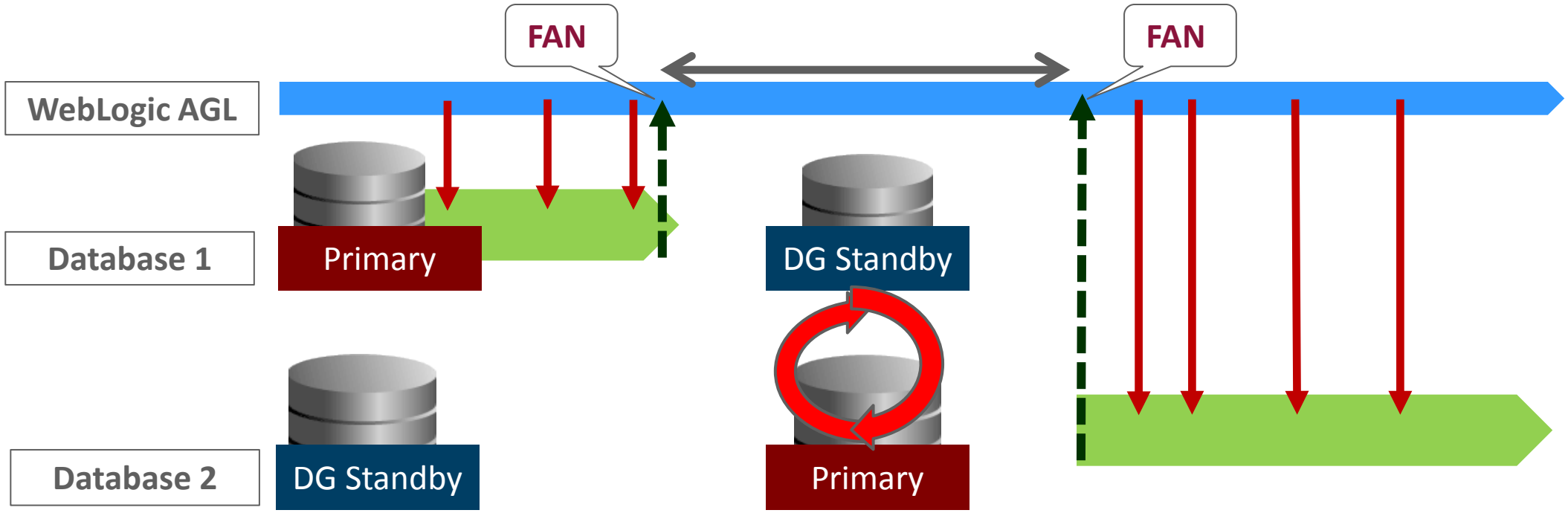
WebLogic Active GridLink and Data Guard

Empowered by Innovation

NEC

1. srvctl stop services on primary site & drain **(e.g. 25s – 30s)**
2. Data Guard switchover
3. New primary database open, start service, rebalance

No errors, application continues



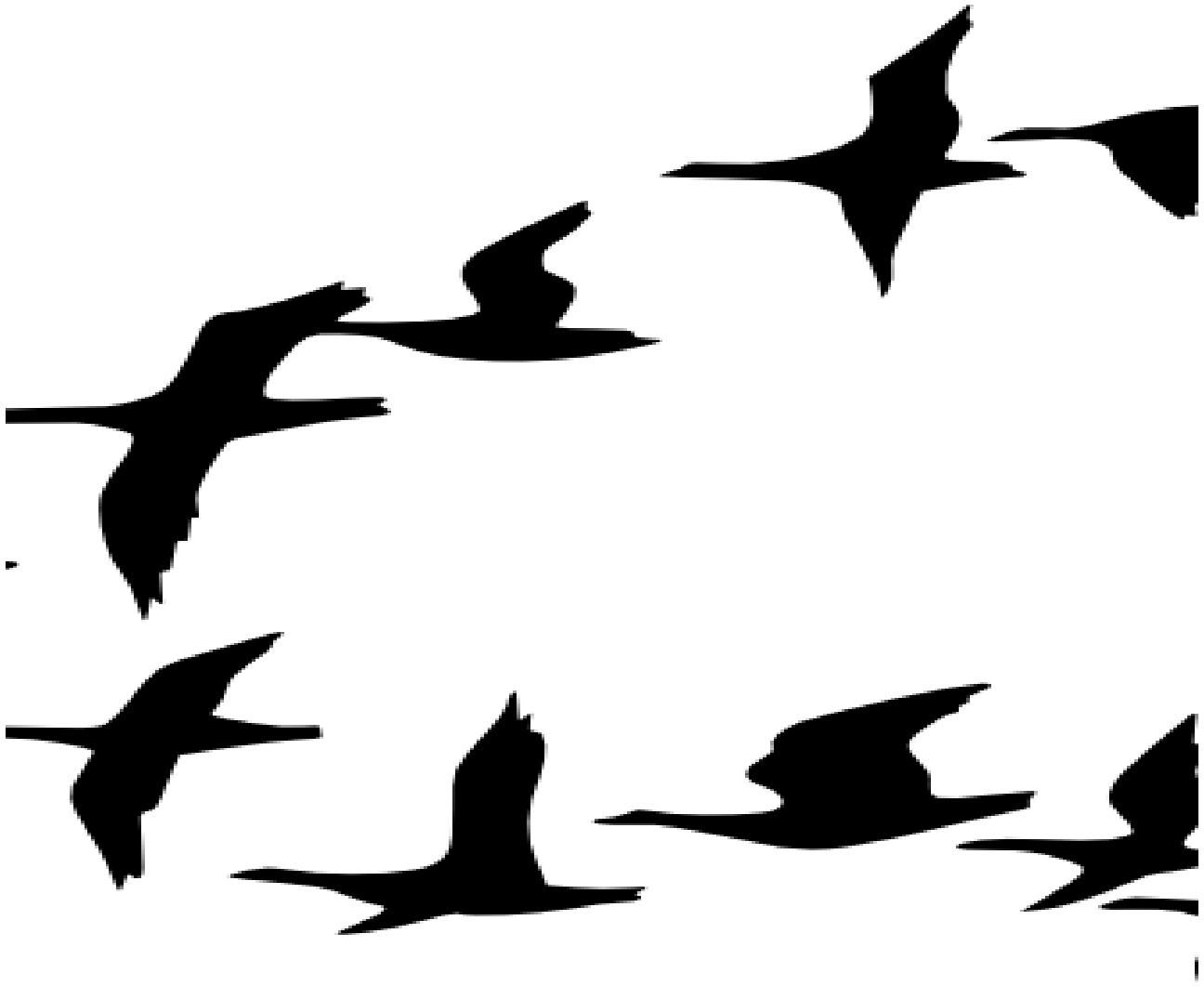
High Availability by Patch Type

	One- Off	PSU/CPU	Bundle Patch	Patch Set
RAC Rolling	96%	All	Most	No
Standby First	98%	All	All	No
Out of Place	All	All	Exadata bundles	No
Online - Hot	82%*	No	No	No

** Available from 11.2.0.2 onward*

Enterprise Applications

Application	Planned maintenance operation	Configuration Change
Siebel	disconnect sessions transactional	NET
PeopleSoft		NET and TAF SELECT
JD Edwards		NET
Informatica		NET



Planned Draining Demonstration

Booth Continuous Availability

SLD-125

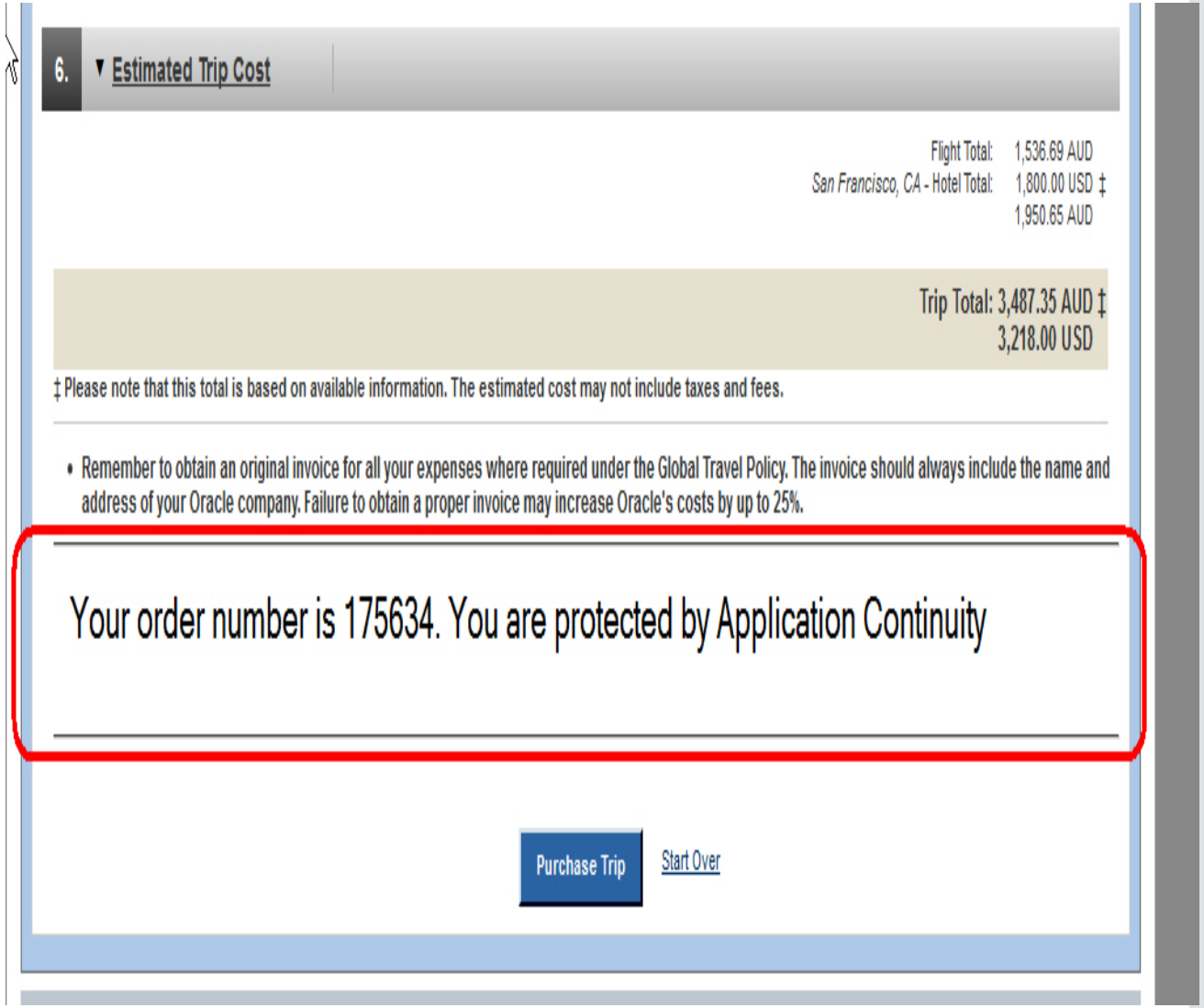
Application Continuity

Unplanned outages should be hidden from applications

Application Continuity

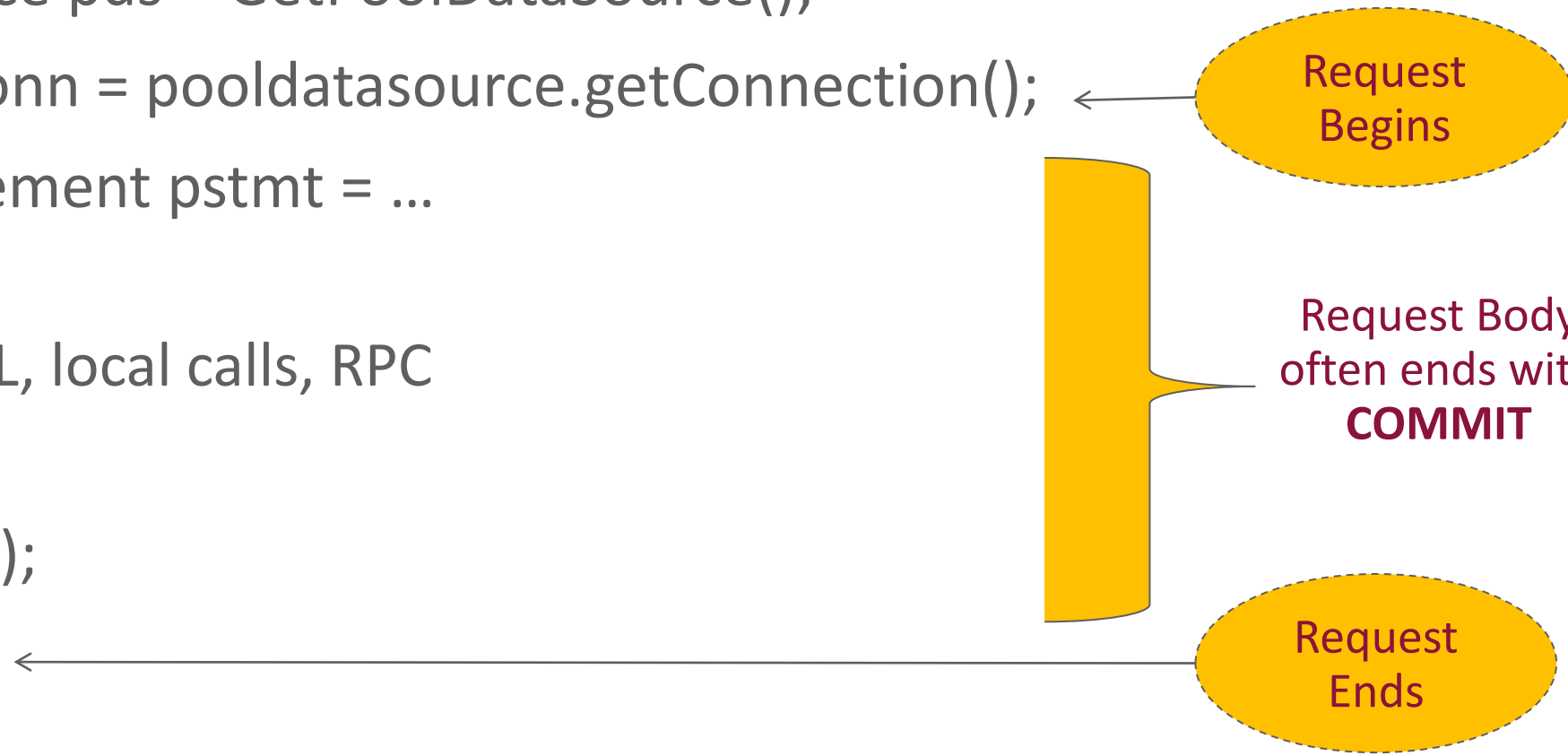
In-flight work continues

- Replays in-flight work on recoverable errors
- Masks most hardware, software, network, storage errors and outages
- Supports JDBC-Thin, UCP, WebLogic Server, 3rd Party Java application servers
- RAC, RAC One, & Active Data Guard
- Improves end user experience



Database Request – UCP example

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



Phases in Application Continuity

1 – Normal Operation

- Marks database requests
- Captures original calls , their inputs, and validation data
- Decides which can and cannot be replayed

2 – Outage Phase 1: Reconnect

- Checks replay is enabled
- Verifies timeliness
- Creates a new connection
- Checks target database is valid
- Uses Transaction Guard to force last outcome

3 – Outage Phase 2: Replay

- Replays captured calls
- Ensures results returned to application match original
- On success, returns control to the application

Steps to use Application Continuity

Check	What to do
Request Boundaries	UCP, WebLogic, and standard 3rd Party App servers – return connections to pool.
JDBC Deprecated Classes	Replace non-standard classes (MOS 1364193.1)
Side Effects	Use disable API if a request has a call that should not be replayed
Callbacks	Register a callback for applications that change state outside requests. For WebLogic and UCP labels – do nothing.
Mutable Functions	Grant keeping mutable values, e.g. sequence.nextval

Disabling Replay

Use disableReplay API for requests that should not be replayed.

Make a conscious decision to replay side effects

e.g. Autonomous Transactions
UTL_HTTP
UTL_URL
UTL_FILE
UTL_FILE_TRANSFER
UTL_SMTP
UTL_TCP
UTL_MAIL
DBMS_JAVA callouts
EXTPROC

Grant Mutables

Keep original function results at replay

ALTER SEQUENCE.. [sequence object] [KEEP|NOKEEP];

CREATE SEQUENCE.. [sequence object] [KEEP|NOKEEP];

For other database users accessing these items :

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

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

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

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

Configuration at Database

Set Service Attributes

FAILOVER_TYPE = TRANSACTION for Application Continuity

Review the service attributes:

- COMMIT_OUTCOME = TRUE for Transaction Guard
- REPLAY_INITIATION_TIMEOUT = 300 after which replay is canceled
- FAILOVER_RETRIES = 30 for the number of connection retries per replay
- FAILOVER_DELAY = 3 for delay in seconds between connection retries

Configuration at Client

Use JDBC Replay Data Source

At WebLogic Console or UCP/Weblogic property file -

```
Select new 12.1 datasource

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

Killing Sessions - Extended

DBA Command	Replays
srvctl stop service -db orcl -instance orcl2 -force	**YES**
srvctl stop service -db orcl -node rws3 -force	**YES**
srvctl stop service -db orcl -instance orcl2 **–noreplay** -force	
srvctl stop service -db orcl -node rws3 **–noreplay** -force	
alter system kill session … immediate	**YES**
alter system kill session … **noreplay**	
dbms_service.disconnect_session([service], dbms_service. **noreplay**)	

Success Stories
Out of the Box

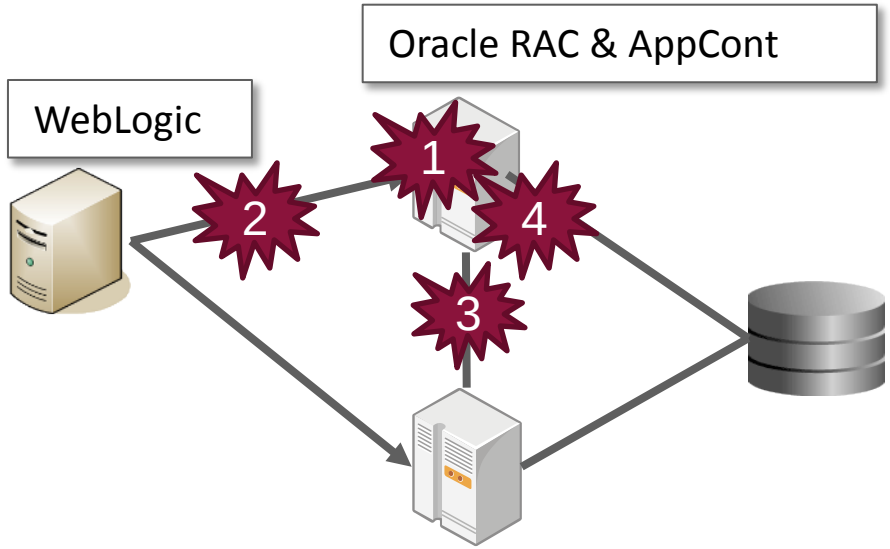
Unplanned Failover with Application Continuity

WebLogic Active GridLink and Real Application Clusters

Empowered by Innovation

NEC

	BEFORE	AFTER
1	DB 11gR2+WLS Generic DS Error AP wait time:1s	DB12c+ GridLink+AppCont No errors, App Continues AP wait time:1s
2	DB 11gR2+WLS Generic DS TIMEOUT 900s (TCP keep-alive)	DB12c+ GridLink+AppCont No errors, App Continues AP wait time:1s
3	DB 11gR2+WLS Generic DS Error AP wait time: 30s	DB12c+ GridLink+AppCont No errors, App Continues AP wait time:30s
4	DB 11gR2+WLS Generic DS Hang AP wait time: minutes	DB12c+ GridLink+AppCont + NEC Monitor : No errors, App Continues



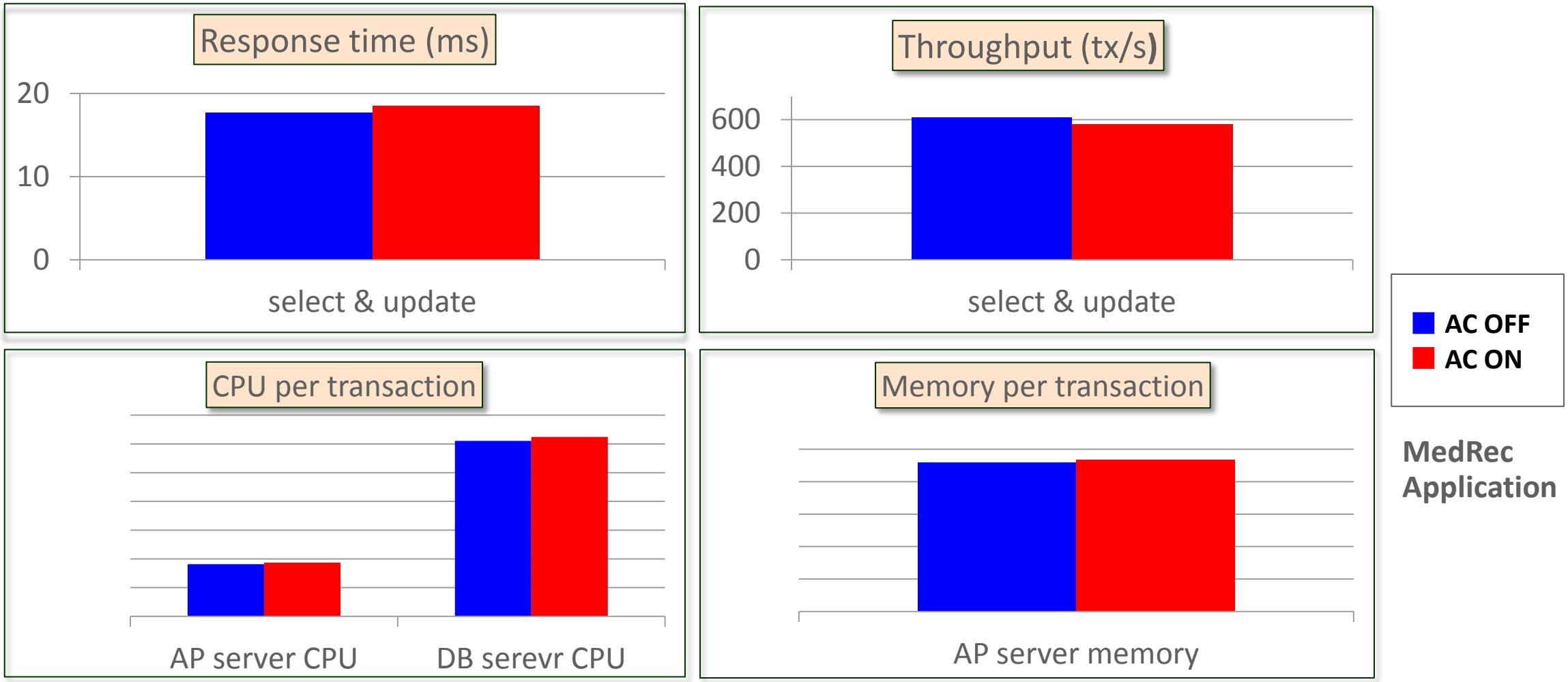
1. Instance down
2. Public network down
3. Interconnect down
4. Background process hang

Application Continuity Performance

WebLogic Server Active GridLink and Real Application Clusters

Empowered by Innovation

NEC



Planned Failover with FAN

WebLogic Server Active GridLink and Real Application Clusters

Empowered by Innovation

NEC

DBA Operation	Maintenance	Result	Time to Drain all Sessions
RAC rolling	PSU apply using opatch	No errors to application	5s
RAC rolling	Instance parameter change	No errors to application	7s
Data Guard switchover	Site maintenance	No errors to application	29s
Data Guard switchover	Site maintenance fallback	No errors to application	25s

Planned and Unplanned Failover

RAC One Node, IBM WebSphere, Universal Connection Pool

Swiss Re

Maintenance	Result	Time allowed
Planned with FAN	No errors to application	4 hours
Unplanned with Application Continuity	No errors to application	10 minutes

ORACLE

Japan Research Institute – Unplanned with AC

WebLogic Server Active GridLink and Real Application Clusters

Workloads	Replay	Reason
concurrent OLTP with DML	succeeds	DML replays concurrently
concurrent OLTP query and DML mix	succeeds	Queries replay at original SCN
concurrent OLTP with select for update and DML mix	most succeed	Rejections only when unable to restore original

Application Continuity Demonstration

Pythian
love your data®

For Developers : Application Continuity offloads the challenging work of transaction resubmission during failure events, allowing developers to focus on functionality.

Christo Kutrovsky – ATCG Principal Consultant, Oracle ACE

For Enterprise Architects : Application Continuity is a major step towards the holy grail of a continuously available, consistent, and highly performing database cluster

Marc Fielding – ATCG Principal Consultant, Oracle

ORACLE

Empowered by Innovation

NEC

The combinatorial solution with Application Continuity, Real Application Clusters, Data Guard, WebLogic Server Active GridLink and NEC hardware and middleware enables us to provide incredibly high available system for our Mission Critical customers. This solution will become our primary solution for cloud and big data areas.

Yuki Moriyama

Senior Manager, NEC Corporation

ORACLE

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.

Hardware and Software
Engineered to Work Together


ORACLE®