19c透明应用连续性(TAC)

(Transparent Application Continuity)

公益讲座11：00准时开始，请大家先浏览云技术微信公众号技术文章。资料会在各群同步发布，已入群客户请勿重复入群!

20-21

数据库和云讲座群

甲骨文云技术公众号

B站专家系列课程

19c 透明应用连续性 (TAC)
(Transparent Application Continuity AC)

甲骨文技术公益课 - 数据库专场

2023 年 5 月 12 日 11:00

线上直播

郭俊龙
CoE(Center of Excellence)

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

目录

1. TAC简介

2. TAC应用

3. TAC限制

4. TAC演示

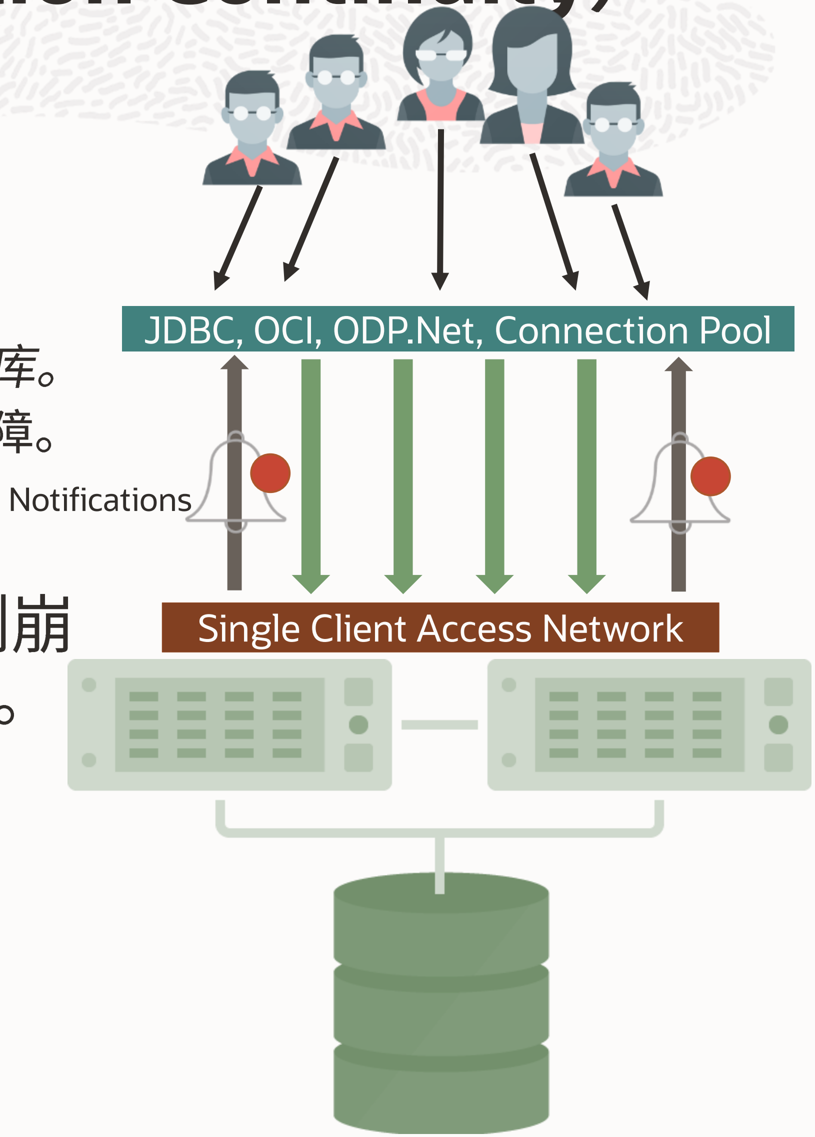
透明应用连续性（Transparent Application Continuity）

什么是TAC?

- 着重于用户体验。
  - ✓ 屏蔽计划内维护和意外停机。
  - ✓ 应用可以透明切换到可用的冗余节点 – *RAC存活节点或者ADG备库。*
  - ✓ TAC通过自动恢复进行中的事务，来为应用程序屏蔽数据库层故障。
  - ✓ 数据库层的中断对用户来说似乎只是稍微延迟的执行。

- 有效地掩盖了如滚动打补丁、网络故障、数据库实例崩溃、以及切换到 Active Data Guard 备库等中断事件。

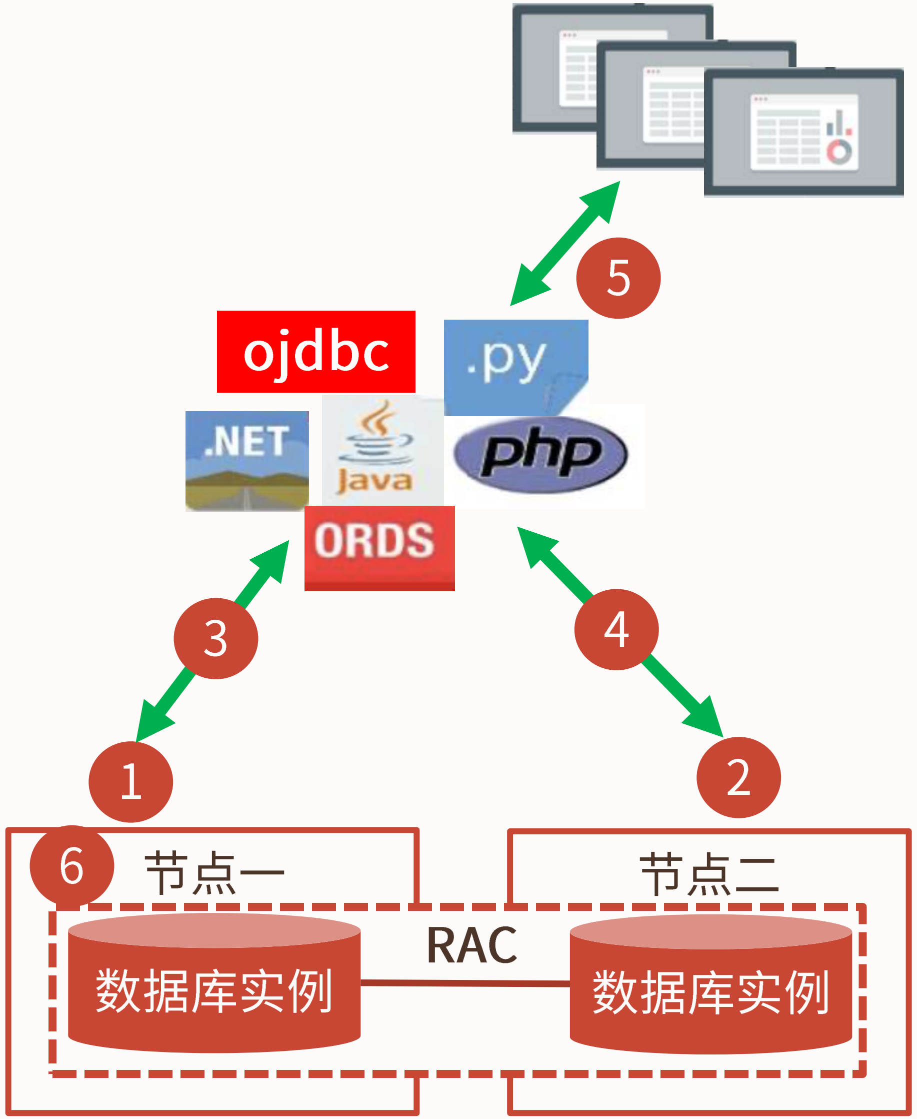
- 配置简单，几乎无代码更改，对应用的透明度高。


JDBC, OCI, ODP.Net, Connection Pool
Notifications
Single Client Access Network

免费咨询热线：400-699-8888

TAC目标是什么？

应对计划内维护和意外停机，为应用层屏蔽掉99%的异常



计划内维护（无停机）：

1 数据库服务已重定位或停止

2 服务在另一个RAC实例上启动

3 连接到服务的会话已排空

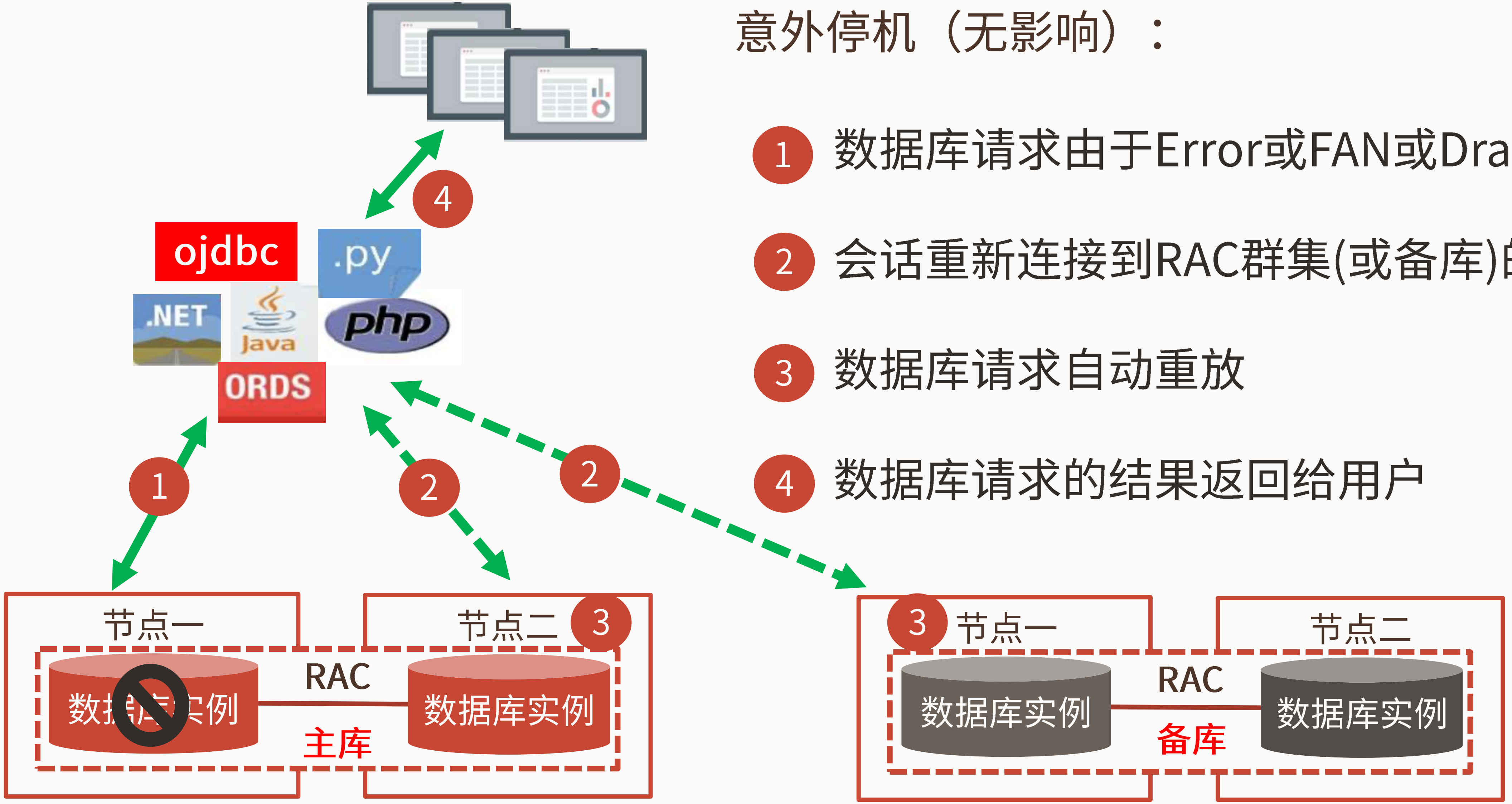
4 新会话连接到另一个实例上的服务

5 数据库请求的结果返回给用户

6 维护活动可以在第一个节点上开始（滚动）

TAC目标是什么？

应对计划内维护和意外停机，为应用层屏蔽掉99%的异常



意外停机（无影响）：

1 数据库请求由于Error或FAN或Drain而中断

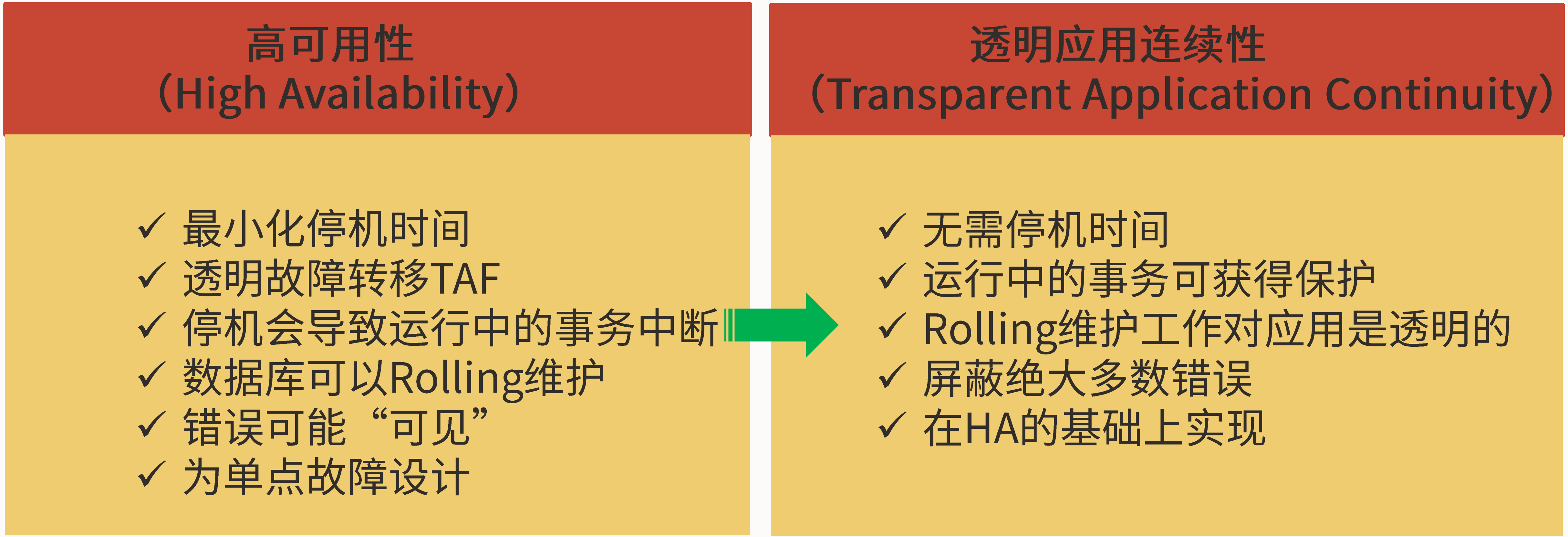
2 会话重新连接到RAC群集(或备库)的可用服务

3 数据库请求自动重放

4 数据库请求的结果返回给用户

从HA到TAC的演进（Evolution ）

从高可用性到透明应用连续性



高可用性（High Availability）

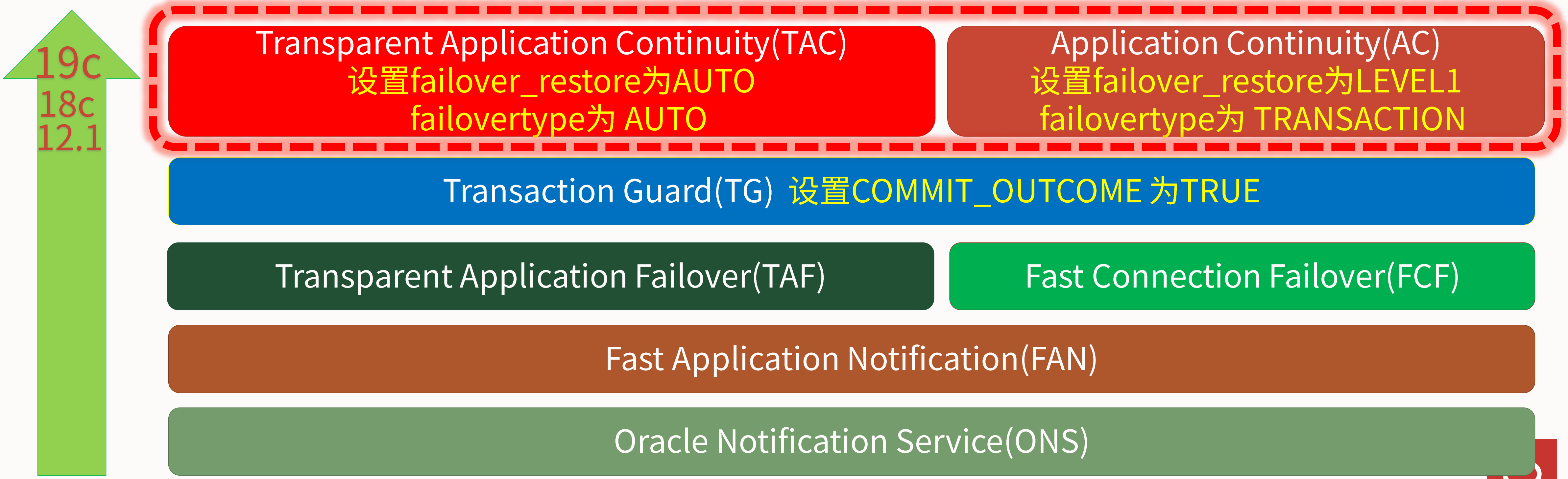
- ✓ 最小化停机时间
- ✓ 透明故障转移TAF
- ✓ 停机会导致运行中的事务中断
- ✓ 数据库可以Rolling维护
- ✓ 错误可能“可见”
- ✓ 为单点故障设计

透明应用连续性（Transparent Application Continuity）

- ✓ 无需停机时间
- ✓ 运行中的事务可获得保护
- ✓ Rolling维护工作对应用是透明的
- ✓ 屏蔽绝大多数错误
- ✓ 在HA的基础上实现

Oracle 19c透明应用连续性（TAC）

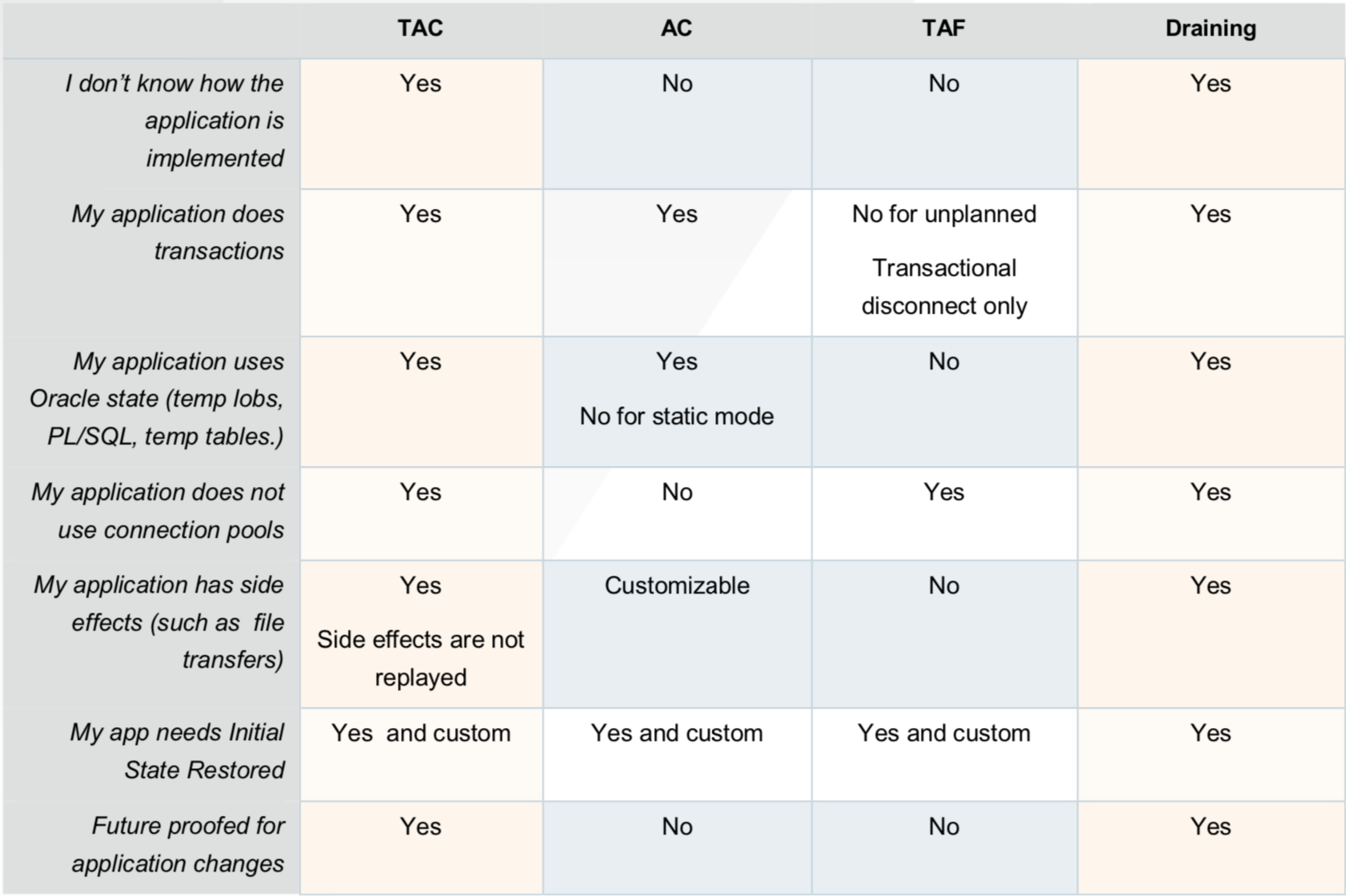
为了提高应用的高可用，Oracle提供了一系列的解决方案，满足各种不同的需求。Oracle引入了事务卫士（Transaction Guard）和应用连续性（Application Continuity），通过这些特性将对客户的影响降至最低。这些本是**应用程序层面**要考虑的问题，现在在**数据库层面**就可以很容易的实现。更重要的是这些功能对于应用来说是**透明的**。涉及到的功能和特性包含有ONS、FAN、TAF 、FCF、TG、AC、TAC等，这些特性在Oracle Database 19c中更易用，他们的逻辑关系如下图：



19c
18c
12.1
Transparent Application Continuity(TAC)
设置failover_restore为AUTO
failovertype为 AUTO
Application Continuity(AC)
设置failover_restore为LEVEL1
failovertype为 TRANSACTION
Transaction Guard(TG) 设置COMMIT_OUTCOME 为TRUE
Transparent Application Failover(TAF)
Fast Connection Failover(FCF)
Fast Application Notification(FAN)
Oracle Notification Service(ONS)

应用连续性

故障转移解决方案



	TAC	AC	TAF	Draining
I don't know how the application is implemented	Yes	No	No	Yes
My application does transactions	Yes	Yes	No for unplanned Transactional disconnect only	Yes
My application uses Oracle state (temp lobs, PL/SQL, temp tables.)	Yes	Yes No for static mode	No	Yes
My application does not use connection pools	Yes	No	Yes	Yes
My application has side effects (such as file transfers)	Yes Side effects are not replayed	Customizable	No	Yes
My app needs Initial State Restored	Yes and custom	Yes and custom	Yes and custom	Yes
Future proofed for application changes	Yes	No	No	Yes

Transparent Application Failover (TAF):

8i, select

12.2: initial session state

Transaction Guard (TG):

12.1, Logical Transaction ID (LTXID)

requires application querying the commit outcome

Application Continuity (AC):

12.2.0.1, OLTP,

recovering the session from a known point which includes session states and transactional states. mutables, e.g. SYSDATE or sequence.NEXTVAL.

Transparent Application Continuity (TAC) & Drainin

18c, achieved by consuming the state-tracking information

--> AC: 12.2, or customize with side effects or callbacks, or complex session states: temporary tables

--> TAF: read only and does not change Oracle session state in the session after the initial setup.

透明应用连续性的阶段

正常操作阶段

- 客户端标记请求
- 客户端捕获原始调用、它们的输入和验证数据
- 服务器端决定哪些可以重放，哪些不能重放，禁用副作用
- 在请求结束时，客户端清除队列，为下一个请求做准备

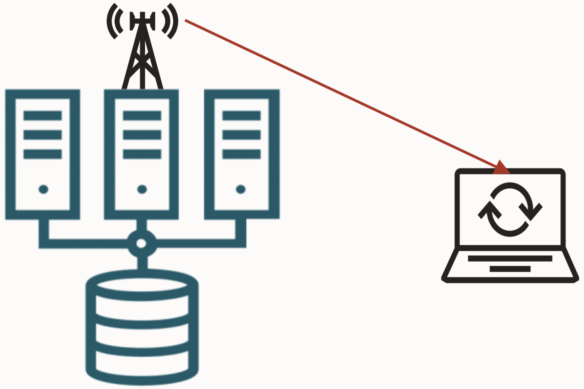
中断重放阶段

- 检查重放已启用
- 创建新连接
- 验证及时性
- 服务器端检查重放是否有效
- 服务器端检查事务是否提交，如果未提交则回滚
- 客户端重放捕获的调用
- 服务器端和客户端确保返回到应用程序的结果与原始结果匹配
- 成功重放后，客户端将控制权返回给应用程序

FAN(Fast Application Notification)与排空(Draining)

快速应用通知 (FAN) 是一种：

- ✓ 高可用性通知机制
- ✓ 通知客户端数据库服务的状态变化
- ✓ 在停机情况下中断会话等待

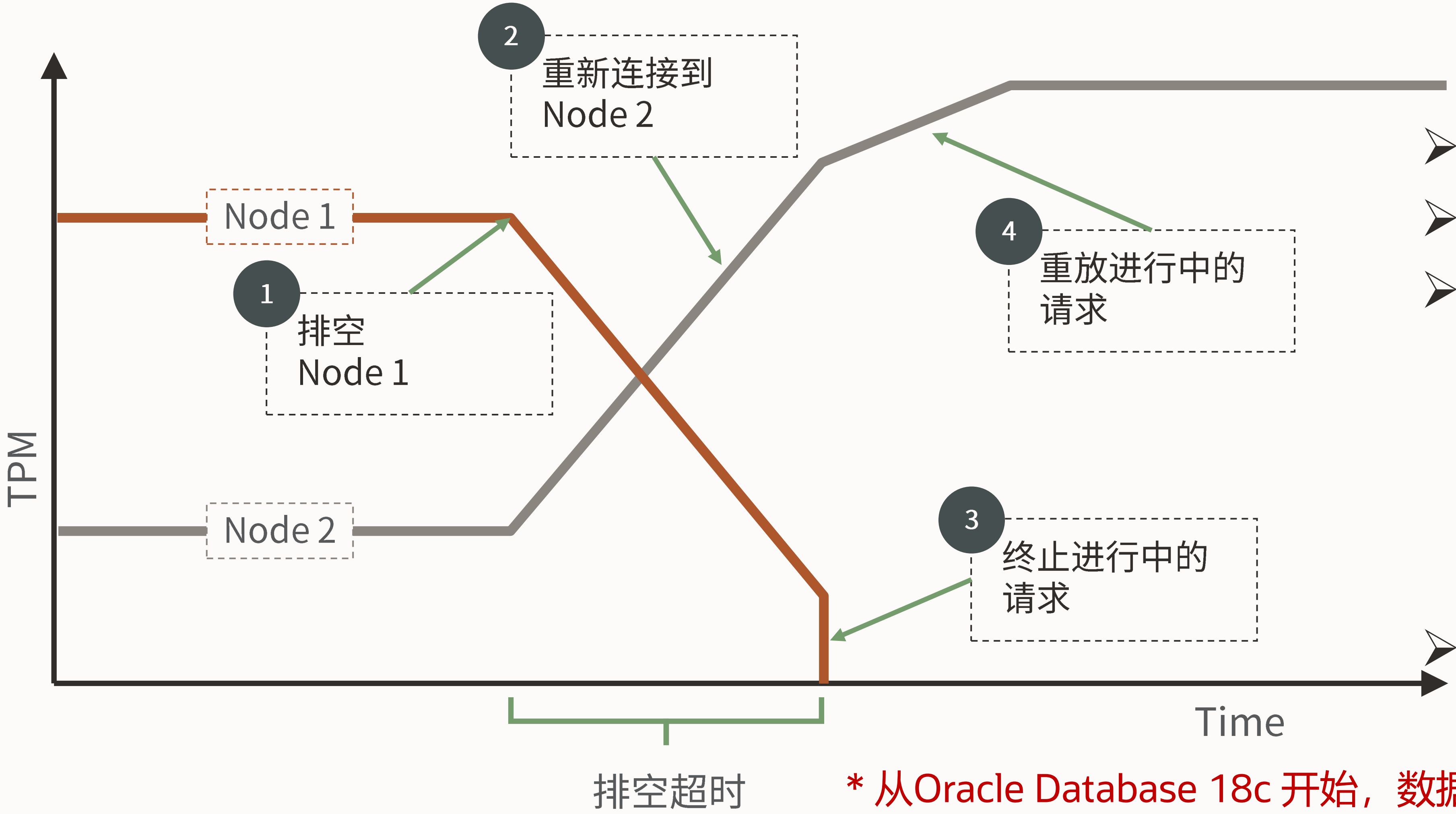


排空（Draining）描述：

- 导致会话在给定节点或实例上完成的一个操作
- 为维护做准备
- Oracle 数据库的一个功能，由 Oracle 驱动程序和连接池以及 SQL 支持

Draining（排空）

Draining,由Oracle Database来排空：排空-> 重新连接-> 故障转移



- ➢ 数据库服务重定位
- ➢ 用户工作自动排空
- ➢ 很多阶段可以排空:
  - ✓归还连接给连接池
  - ✓连接测试
  - ✓注销
  - ✓会话迁移
- ➢ 计划的故障转移

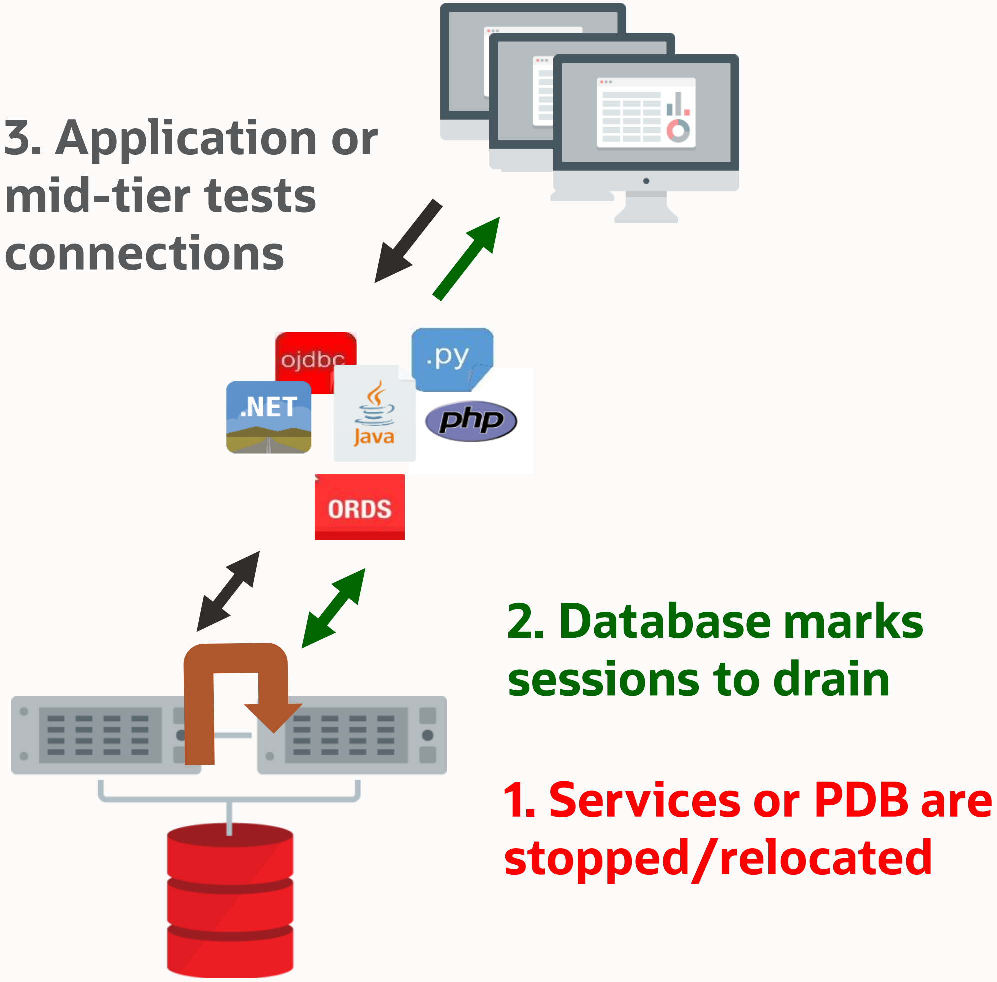
* 从Oracle Database 18c 开始，数据库本身会排空会话。

免费咨询热线：400-699-8888

启用连接测试来排空

19c Database and 19c Drivers, and newer

NEW IN 19c



3. Application or mid-tier tests connections
ojdbc
.py
.NET
Java
php
ORDS
2. Database marks sessions to drain
1. Services or PDB are stopped/relocated

- 应用程序“测试”连接
- 数据库或驱动响应连接错误
- 新的工作在另一个新连接上继续

View in DBA_CONNECTION_TESTS add more tests with DBMS_APP_CONT_ADMIN

启用连接测试

➢ 客户端侧（驱动程序连接测试）：

✓ **UCP** use one of the following tests, isValid() is the preferred method:

java.sql.Connection.isValid(int timeout) or

oracle.jdbc.OracleConnection.pingDatabase() or

oracle.jdbc.OracleConnection.pingDatabase(int timeout) or

a HINT at the start of your test SQL: /*+ CLIENT_CONNECTION_VALIDATION */

✓ **OCI** use OCI_ATTR_SERVER_STATUS

✓ **ODP.NET** CheckConStatus is on by default.

➢ 数据库侧（SQL连接测试）：

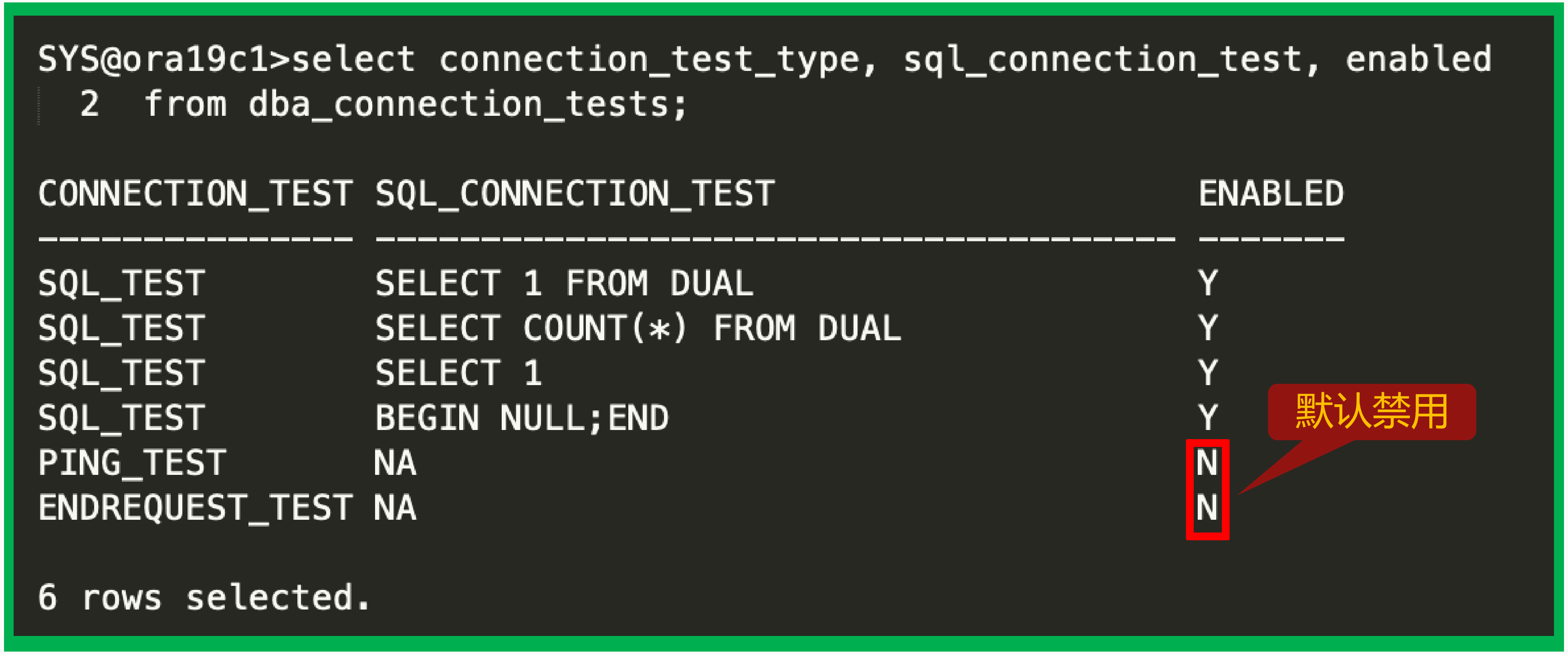
登录到数据库查看dba_connection_tests视图:

SQL>select connection_test_type, sql_connection_test, enabled from dba_connection_tests;

DBA_CONNECTION_TESTS查看添加和启用的连接测试:

Database Side ONLY. CDB, PDB, Service Levels

SQL>select connection_test_type, sql_connection_test, enabled from dba_connection_tests;



```
SYS@ora19c1>select connection_test_type, sql_connection_test, enabled
  2  from dba_connection_tests;

CONNECTION_TEST SQL_CONNECTION_TEST                      ENABLED
--------------- ---------------------------------------- -------
SQL_TEST        SELECT 1 FROM DUAL                       Y
SQL_TEST        SELECT COUNT(*) FROM DUAL                Y
SQL_TEST        SELECT 1                                 Y
SQL_TEST        BEGIN NULL;END                           Y
PING_TEST       NA                                       N
ENDREQUEST_TEST NA                                       N

6 rows selected.
```

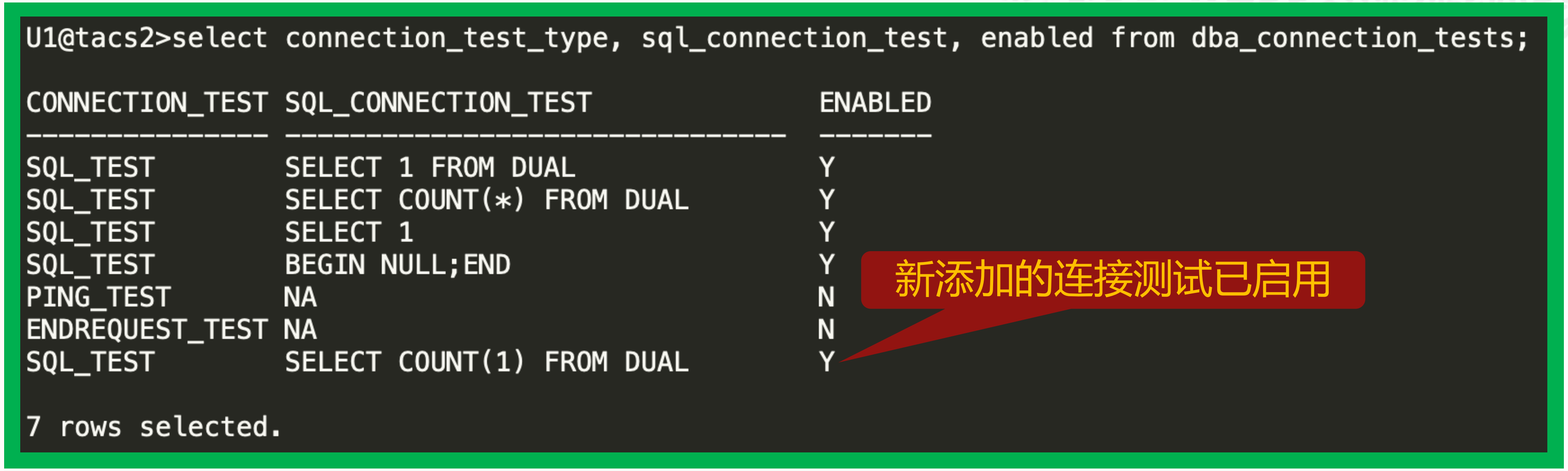
默认禁用

DBMS_APP_CONT_ADMIN管理数据库的连接测试:

使用**dbms_app_cont_admin**添加、删除、启用或禁用服务、可插入数据库或非容器数据库的连接测试:

- 添加新的连接测试:

  SQL> EXECUTE dbms_app_cont_admin.**add**_sql_connection_test('SELECT COUNT(1) FROM DUAL');



```
U1@tacs2>select connection_test_type, sql_connection_test, enabled from dba_connection_tests;

CONNECTION_TEST SQL_CONNECTION_TEST              ENABLED
--------------- -------------------------------- -------
SQL_TEST        SELECT 1 FROM DUAL               Y
SQL_TEST        SELECT COUNT(*) FROM DUAL        Y
SQL_TEST        SELECT 1                         Y
SQL_TEST        BEGIN NULL;END                   Y
PING_TEST       NA                               N
ENDREQUEST_TEST NA                               N
SQL_TEST        SELECT COUNT(1) FROM DUAL        Y

7 rows selected.
```

新添加的连接测试已启用

- 禁用新的连接测试:

  SQL> EXECUTE dbms_app_cont_admin.**disable**_connection_test(dbms_app_cont_admin.sql_test,'SELECT COUNT(1) FROM DUAL');

- 启用新的连接测试:

  SQL> EXECUTE dbms_app_cont_admin.**enable**_connection_test(dbms_app_cont_admin.sql_test,'SELECT COUNT(1) FROM DUAL');

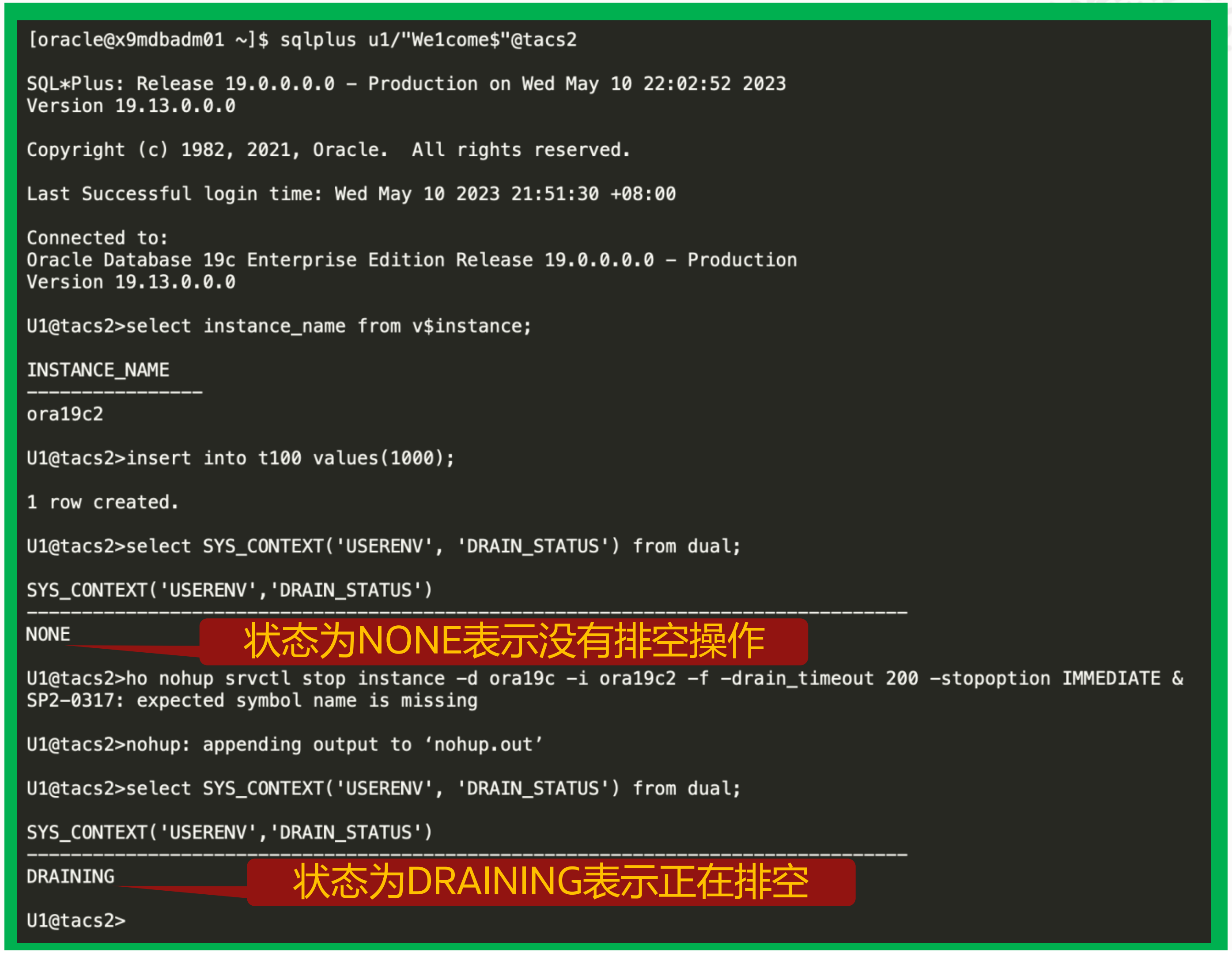
- 删除新的连接测试:

  SQL> EXECUTE dbms_app_cont_admin.**delete**_sql_connection_test('SELECT COUNT(1) FROM DUAL');

检查排空状态：

检查排空状态，使用函数**userenv**来确定会话是否处于排空模式：

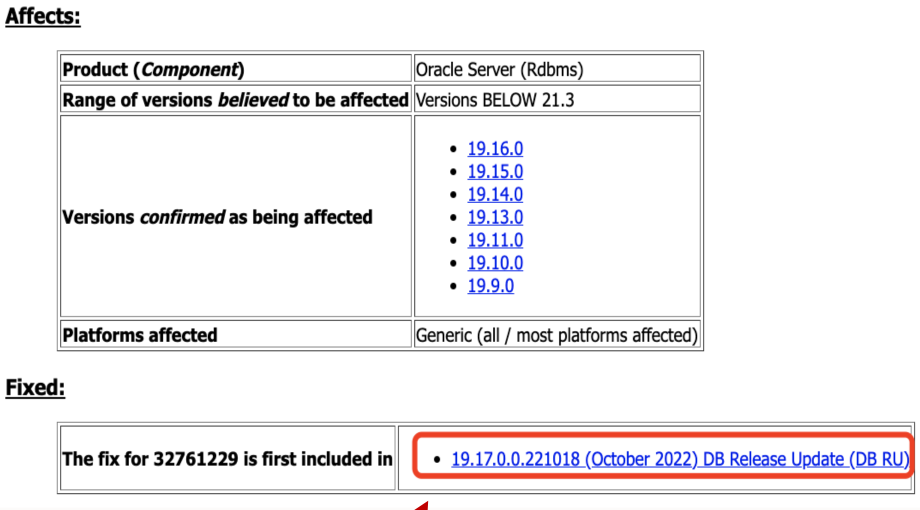
SQL> select SYS_CONTEXT('USERENV', 'DRAIN_STATUS') from dual ;



```
[oracle@x9mdbadm01 ~]$ sqlplus u1/"We1come$"@tacs2

SQL*Plus: Release 19.0.0.0.0 - Production on Wed May 10 22:02:52 2023
Version 19.13.0.0.0

Copyright (c) 1982, 2021, Oracle.  All rights reserved.

Last Successful login time: Wed May 10 2023 21:51:30 +08:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.13.0.0.0

U1@tacs2>select instance_name from v$instance;

INSTANCE_NAME
----------------
ora19c2

U1@tacs2>insert into t100 values(1000);

1 row created.

U1@tacs2>select SYS_CONTEXT('USERENV', 'DRAIN_STATUS') from dual;

SYS_CONTEXT('USERENV','DRAIN_STATUS')
--------------------------------------------------------------------------------
NONE

U1@tacs2>ho nohup srvctl stop instance -d ora19c -i ora19c2 -f -drain_timeout 200 -stopoption IMMEDIATE &
SP2-0317: expected symbol name is missing

U1@tacs2>nohup: appending output to ‘nohup.out’

U1@tacs2>select SYS_CONTEXT('USERENV', 'DRAIN_STATUS') from dual;

SYS_CONTEXT('USERENV','DRAIN_STATUS')
--------------------------------------------------------------------------------
DRAINING

U1@tacs2>
```

状态为NONE表示没有排空操作

状态为DRAINING表示正在排空



Affects:

Product (*Component*)	Oracle Server (Rdbms)
Range of versions *believed* to be affected	Versions BELOW 21.3
Versions *confirmed* as being affected	• 19.16.0 • 19.15.0 • 19.14.0 • 19.13.0 • 19.11.0 • 19.10.0 • 19.9.0
Platforms affected	Generic (all / most platforms affected)

Fixed:

The fix for 32761229 is first included in	• 19.17.0.0.221018 (October 2022) DB Release Update (DB RU)

☆ **Bug 32761229 - Includes SYS_CONTEXT('USERENV', 'DRAIN_STATUS') in 19c to see draining status of sessions (Doc ID 32761229.8)**

为应用程序服务器启用连接测试（1)

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)	validate-on-match Check-valid-connection-sql	SQL – SELECT COUNT(*) FROM DUAL
Apache Tomcat	TestOnBorrow TestOnRelease	SQL – SELECT 1 FROM DUAL
ODP.Net Unmanaged	Connection.status()	OCI_ATTR_SERVER_STATUS

为应用程序服务器启用连接测试（2）

Application	Condition	Connection Test to DB
eBusiness Suite	Connection borrowed from WebLogic	TestConnectionsOnReserve with “BEGIN NULL;END;”
Fusion Applications	Connection returned to WebLogic and C++ pools and checked	TestConnectionsOnReserve with isValid() OCI_ATTR_SERVER_STATUS
Siebel	Connection requested	OCI_ATTR_SERVER_STATUS
Peoplesoft	Connection requested	OCI_ATTR_SERVER_STATUS
Customer example	Custom pool with Metadata table Checks status every 60 seconds	OCI_ATTR_SERVER_STATUS

Session状态恢复

TAC（Transparent Application Continuity）会透明地跟踪和记录会话和事务状态，以便在可恢复的中断之后恢复数据库会话在重放之前确保会话状态是正确的非常重要。

会话状态示例：NLS settings, optimizer preferences, event settings, PL/SQL global variables, temporary tables, advanced queues, LOBs, and result cache.

nls_nchar_conv_excp
nls_calendar
nls_comp
nls_currency
nls_date_format
nls_date_language
nls_dual_currency
nls_iso_currency
nls_language
nls_length_semantics
nls_numeric_characters
nls_sort
nls_territory
nls_time_format
nls_time_tz_format
nls_timestamp_format
nls_timestamp_tz_format
allow_rowid_column_type
approx_for_aggregation

approx_for_count_distinct
approx_for_percentile
cursor_sharing
default_collation
optimizer_capture_sql_plan_baselines
optimizer_ignore_hints
optimizer_ignore_parallel_hint
optimizer_use_sql_plan_baselines
parallel_degree_limit
parallel_degree_policy
parallel_min_time_threshold
plscope_settings
plsql_ccflags
plsql_debug
plsql_optimize_level
plsql_warnings
recyclebin
result_cache_mode

aq_tm_processes
cell_offload_compaction
cell_offload_parameters
cell_offload_plan_display
cell_offload_processing
cell_offloadgroup_name
commit_logging
commit_wait
commit_write
containers_parallel_degree
create_stored_outlines
cursor_bind_capture_destination
cursor_invalidation
db_index_compression_inheritance
db_securefile
db_unrecoverable_scn_tracking
ddl_lock_timeout
deferred_segment_creation
dst_upgrade_insert_conv
global_names
hash_area_size
heat_map
inmemory_clause_default
inmemory_query
java_jit_enabled
max_dump_file_size
multishard_query_data_consistency
multishard_query_partial_results
object_cache_max_size_percent
object_cache_optimal_size
olap_page_pool_size
optimizer_adaptive_plans

optimizer_adaptive_reporting_only
optimizer_adaptive_statistics
optimizer_dynamic_sampling
optimizer_features_enable
optimizer_index_caching
optimizer_index_cost_adj
optimizer_inmemory_aware
optimizer_mode
optimizer_use_invisible_indexes
optimizer_use_pending_statistics
parallel_force_local
parallel_instance_group
parallel_min_degree
parallel_min_percent
plsql_code_type
plsql_v2_compatibility
query_rewrite_enabled
query_rewrite_integrity
remote_dependencies_mode
result_cache_remote_expiration
resumable_timeout
session_cached_cursors
skip_unusable_indexes
smtp_out_server
sort_area_retained_size
sort_area_size
spatial_vector_acceleration
sql_trace
sqltune_category
star_transformation_enabled
statistics_level
temp_undo_enabled
timed_os_statistics
timed_statistics
tracefile_identifier
workarea_size_policy

NEW IN 20c

KILL & DISCONNECT Session注意事项

如果配置了“应用程序连续性”，并且DBA通过使用ALTER SYSTEM KILL session或ALTER SYSTEM DISCONNECT session语句终止或断开会话，则默认情况下，“应用程序持续性”会尝试恢复会话。但是，如果不希望重放会话，请加上NOREPLAY关键字。

DBA Command	Replays
alter system kill session '*sid, serial#, @inst*' immediate;	**YES**
alter system kill session '*sid, serial#, @inst*' **noreplay**;	**NO**
alter system disconnect session '*sid, serial#, @inst*' immediate;	**YES**
alter system disconnect session '*sid, serial#, @inst*' **noreplay;**	**NO**
srvctl stop service -db orcl -instance orcl2 -force	**YES**
srvctl stop service -db orcl -instance orcl2 -force **-noreplay**	**NO**
srvctl stop service -db orcl -node mynode3 -force	**YES**
srvctl stop service -db orcl -node mynode3 -force **-noreplay**	**NO**
srvctl stop instance -node mynode3 -force	**YES**
srvctl stop instance -node mynode3 -force **-noreplay**	**NO**
dbms_service.disconnect_session([service], dbms_service. **noreplay**)	**NO**

支持的高可用性部署

	Fast Application Notification (FAN)	Runtime Load Balancing (RLB)	Transparent Application Failover (TAF)	Transaction Guard (TG)	Application Continuity (AC)	Transparent Application Continuity (TAC)
Real Application Clusters (RAC, RAC One)	✓	✓ (RAC)	✓	✓	✓	✓
Data Guard (DG) physical standby	With clusterware	x	✓	✓	x	x
Active Data Guard (ADG)	With clusterware	x	✓	✓	✓	✓
RAC+DG (physical standby)	✓	✓ (within RAC)	✓	✓	✓	✓
Global Data Services (GDS)	✓	✓	✓	✓	✓	✓
Golden Gate	x	x	✓ (no DML retry)	x	x	x

目录

1. TAC简介

2. TAC应用

3. TAC限制

4. TAC演示

部署您的应用程序

使用TAC步骤:

1. **配置应用程序**
   - ✓ 使用推荐的连接串
   - ✓ 使用支持的客户端版本
   - ✓ 应用程序满足条件

2. **为您的工作负载配置数据库端服务(DBA)**
   - ✓ 正确配置的数据库和集群架构
   - ✓ 为特定工作负载创建数据库服务
   - ✓ 确保连接字符串使用此数据库服务

3. **为您应用程序使用的可变函数授权(DBA)**
   - ✓ 识别应用程序中使用的可变函数，DBA授权正确权限

4. **测试和评估重放保护**
   - ✓ 检查相关视图、AWR以获取保护的统计信息
   - ✓ 使用ACCHK实用程序识别重放异常

使用推荐的连接串：

将连接串与内置的超时、重试和延迟一起使用，以便传入的连接在中断期间不会看到错误。

Use this Connection String for ALL Oracle driver version 12.2 or higher:

RAC环境：

```
Alias (or URL) =
 (DESCRIPTION =
  (CONNECT_TIMEOUT= 90)(RETRY_COUNT=50)(RETRY_DELAY=3)(TRANSPORT_CONNECT_TIMEOUT=3)
  (ADDRESS_LIST = (LOAD_BALANCE=on)(ADDRESS = (PROTOCOL = TCP)(HOST=primary-scan)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME = YOUR SERVICE)))
```

RAC+ADG环境：

```
Alias (or URL) =
 (DESCRIPTION =
  (CONNECT_TIMEOUT= 90)(RETRY_COUNT=50)(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=standby-scan)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME = YOUR SERVICE)))
```

使用支持的客户端版本：

TAC支持的客户端：

- ✓ Oracle JDBC Replay Driver 18c or later. This is a JDBC driver feature provided with Oracle Database 18c for Application Continuity
- ✓ Oracle Universal Connection Pool (UCP) 18c or later with Oracle JDBC Replay Driver 18c or later.
- ✓ Oracle WebLogic Server 18c, or third-party JDBC application servers using UCP with Oracle JDBC Replay Driver 18c or later
- ✓ Java connection pools or standalone Java applications using Oracle JDBC Replay Driver 18c or later with Request Boundaries
- ✓ OCI Session Pool 18c or later
- ✓ SQL*Plus 18c or later
- ✓ ODP.NET pooled, Unmanaged Driver 18c or later (“Pooling=true” default in 12.2 and later)
- ✓ OCI-based applications using 19c OCI driver or later

应用程序满足条件（1）：

使用快速应用程序通知(FAN)：--Oracle10.2开始支持FAN

- ✓ FAN在服务中断或恢复时立即通知应用程序（要求打开ONS端口，默认6200）。
- ✓ 没有FAN，如果在硬件和网络故障时，应用程序会挂起出现TCP/IP超时，并且在资源恢复时忽略重新平衡。
- ✓ 所有Oracle pool和Oracle应用程序服务器都使用FAN。
- ✓ 第三方JAVA应用服务器可以使用UCP来启用FAN。

FAN已经整合到了下面的产品中：

- Oracle Fusion Middleware and Oracle WebLogic Server
- Oracle Data Guard Broker
- Oracle JDBC Universal Connection Pool or Driver for both JDBC thin and OCI interfaces
- ODP.NET Connection Pool for Unmanaged and Managed Providers
- Oracle Tuxedo
- SQL*Plus
- PHP
- Global Data Services
- Third party JDBC application servers using Oracle JDBC Universal Connection Pool
- Listeners

应用程序满足条件（2）：

根据客户端的不同，在应用程序配置属性中启用FAN（FCF），如下所示

- ✓ **Universal Connection Pool**

  Set the property FastConnectionFailoverEnabled
- ✓ **WebLogic Active GridLink for Oracle RAC**

  FAN and Fast Connection Failover are enabled by default
- ✓ **IBM WebSphere, IBM Liberty, Apache Tomcat, Red Hat WildFly (WildFly (JBoss)), third-party Application Servers**

  Use Universal Connection Pool as a connection pool replacement
- ✓ **ODP.Net clients (Managed and Unmanaged Providers)**

  Set “HA events = true;pooling=true” in the connect string if using ODP.Net 12.1 or earlier
- ✓ **OCI clients**

  OCI clients using oraaccess.xml set events to true
- ✓ **SQL*Plus enables FAN by default**
- ✓ **PHP**

  In php.ini add the entry oci8.events=on
- ✓ **Python pool creation**

  cx_Oracle.SessionPool(…,events = True)
- ✓ **Node.js pool creation**

  oracledb.events = true;

应用程序满足条件（3）：

FAN它是如何工作的?

1. 数据库服务状态更改时触发FAN事件;

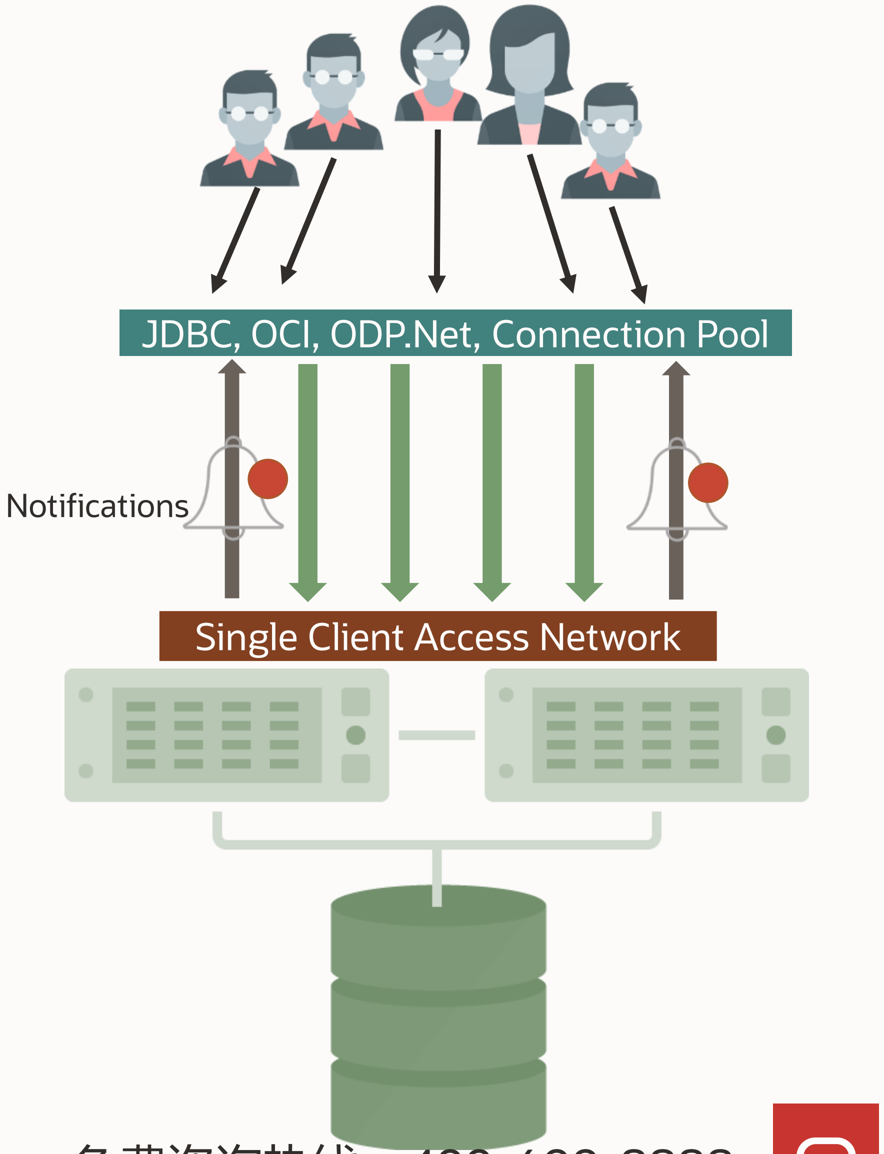
2. FAN事件通过Oracle通知服务（ONS）进行广播;

3. 广播被所有注册的终端（应用程序）接收;

4. 通知的应用程序可以快速响应这些更改，而不是等待超时等。

FAN事件：

- START
- REBALANCE
- PLANNED DOWN ← planned maintenance initiation
- HARD DOWN
- …


JDBC, OCI, ODP.Net, Connection Pool
Notifications
Single Client Access Network

应用程序满足条件（4）：

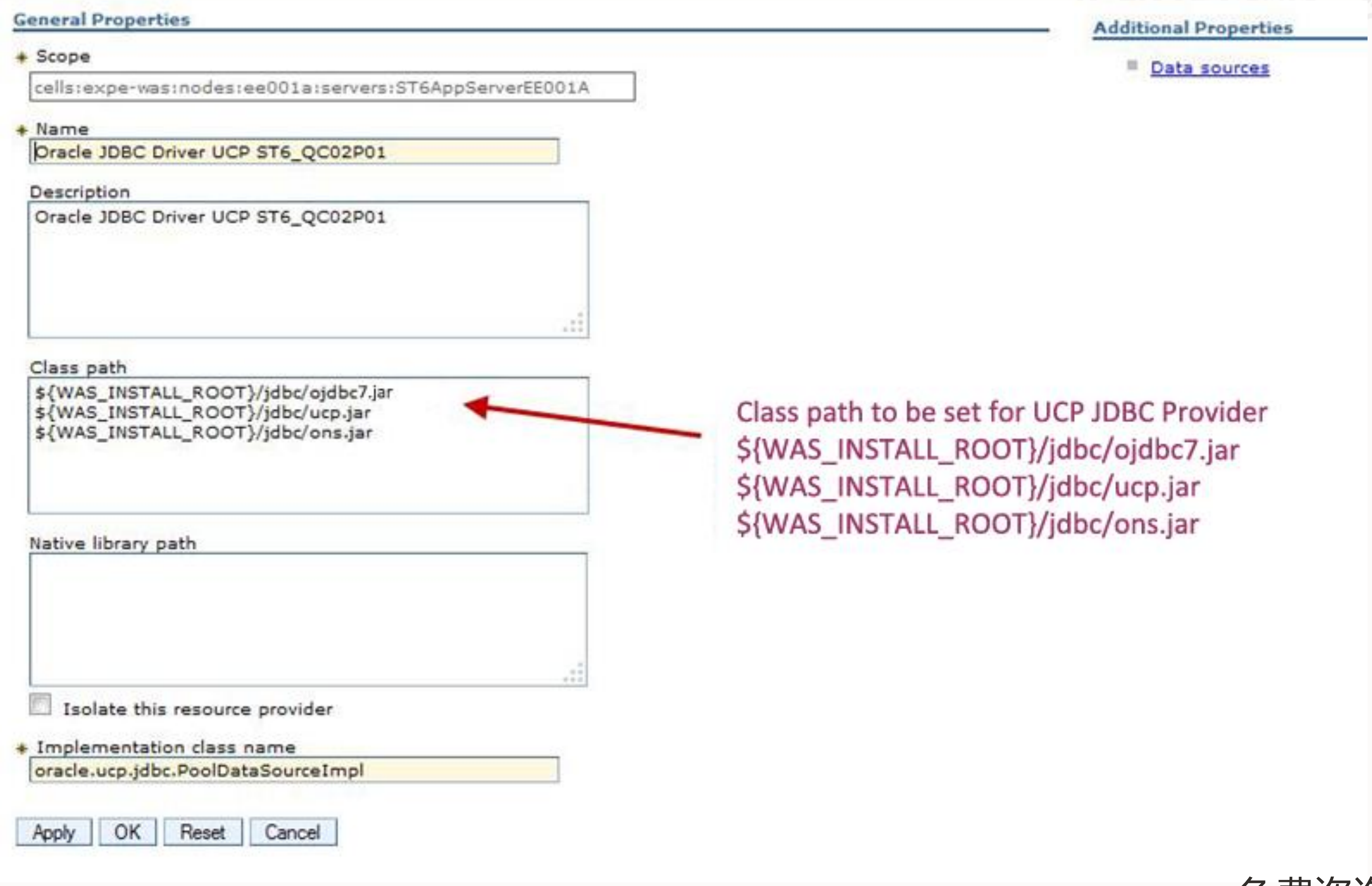
使用推荐的应用程序实践

- ✓ 应用程序使用的最佳实践是在需要时从连接池中获取连接，然后在当前操作完成时将其归还到连接池中。
- ✓ Oracle建议使用支持FAN的Oracle连接池来屏蔽计划内的维护。当应用程序使用带有FAN的Oracle连接池时，在请求之间将连接返回到池时，对用户没有影响。
- ✓ 当Oracle连接池接收到计划停机的FAN事件时，它会将实例上的所有连接标记为排空。立即关闭已归还的连接，以便不再使用它们。当正在使用的连接返回到池时，它们将被关闭。随着时间的推移会优雅地关闭所有连接。
- ✓ 如果您使用的是基于Java的第三方的Application Server，那么实现排空会话和故障转移的最有效方法是，用UCP替换现有的数据源连接。许多应用服务器都支持这种方法，包括IBM WebSphere、IBM Liberty、Apache Tomcat、Red Hat WildFly（WildFly（JBoss））、Spring、Hibernate和其他。Oracle和其他提供商（如IBM）的白皮书描述了如何在这些Application Server上使用UCP。使用UCP作为数据源，可以使用很多UCP的功能，如快速连接故障转移、运行时负载均衡、应用程序连续性和透明的应用程序连续性。
- ✓ 使用连接测试（connection tests ）：分为客户端侧（驱动程序）连接测试 、数据库侧(SQL连接测试)。

免费咨询热线：400-699-8888

UCP与其他基于Java的应用程序服务器

A simple data source replacement



General Properties

* Scope

cells:expe-was:nodes:ee001a:servers:ST6AppServerEE001A

* Name

Oracle JDBC Driver UCP ST6_QC02P01

Description

Oracle JDBC Driver UCP ST6_QC02P01

Class path

```
${WAS_INSTALL_ROOT}/jdbc/ojdbc7.jar
${WAS_INSTALL_ROOT}/jdbc/ucp.jar
${WAS_INSTALL_ROOT}/jdbc/ons.jar
```

Native library path

□ Isolate this resource provider

* Implementation class name

oracle.ucp.jdbc.PoolDataSourceImpl

Additional Properties

- Data sources

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

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

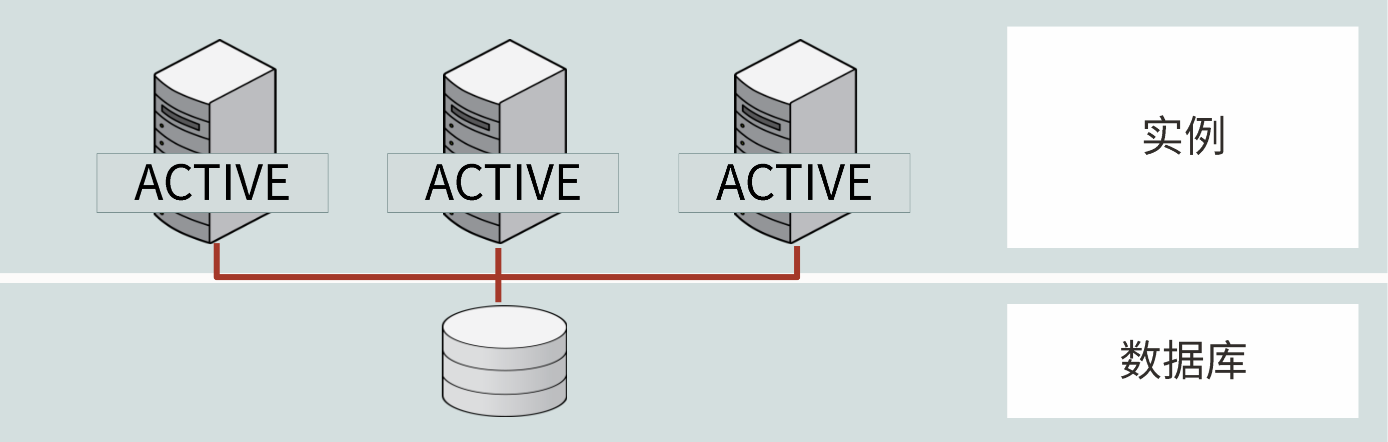
Tip: Excludes TP Managed 2PC XA

正确配置的数据库和集群架构

根据Oracle MAA最佳实践进行配置-数据库层高可用方案

Real Application Cluster （RAC）

- 使用多个实例操作一个数据库的功能
- 所有节点都处于活动状态的共享磁盘类型的群集数据库



Data Guard

- 将主数据库重做日志（Redo）传输到容灾端的备用数据库，然后在备用数据库中应用重做日志以保持数据和主数据库同步。
- 可以切换到备库。

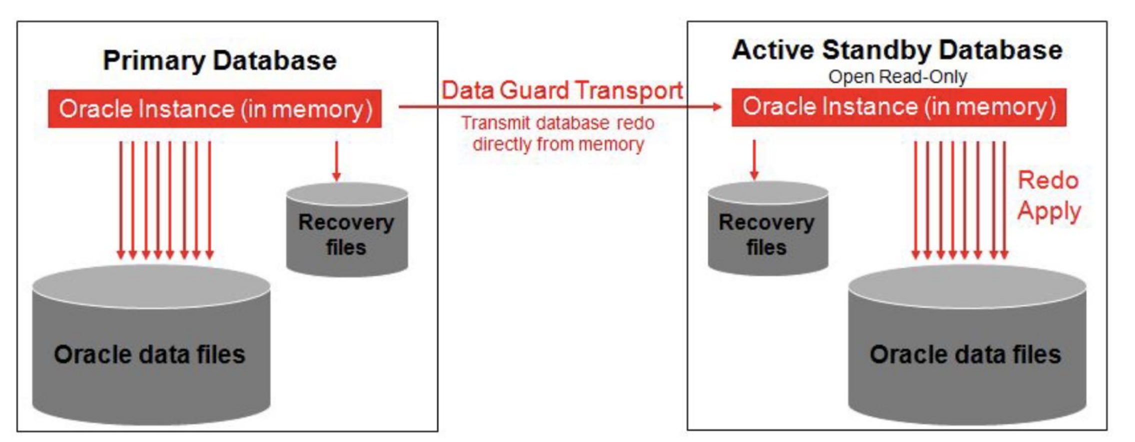
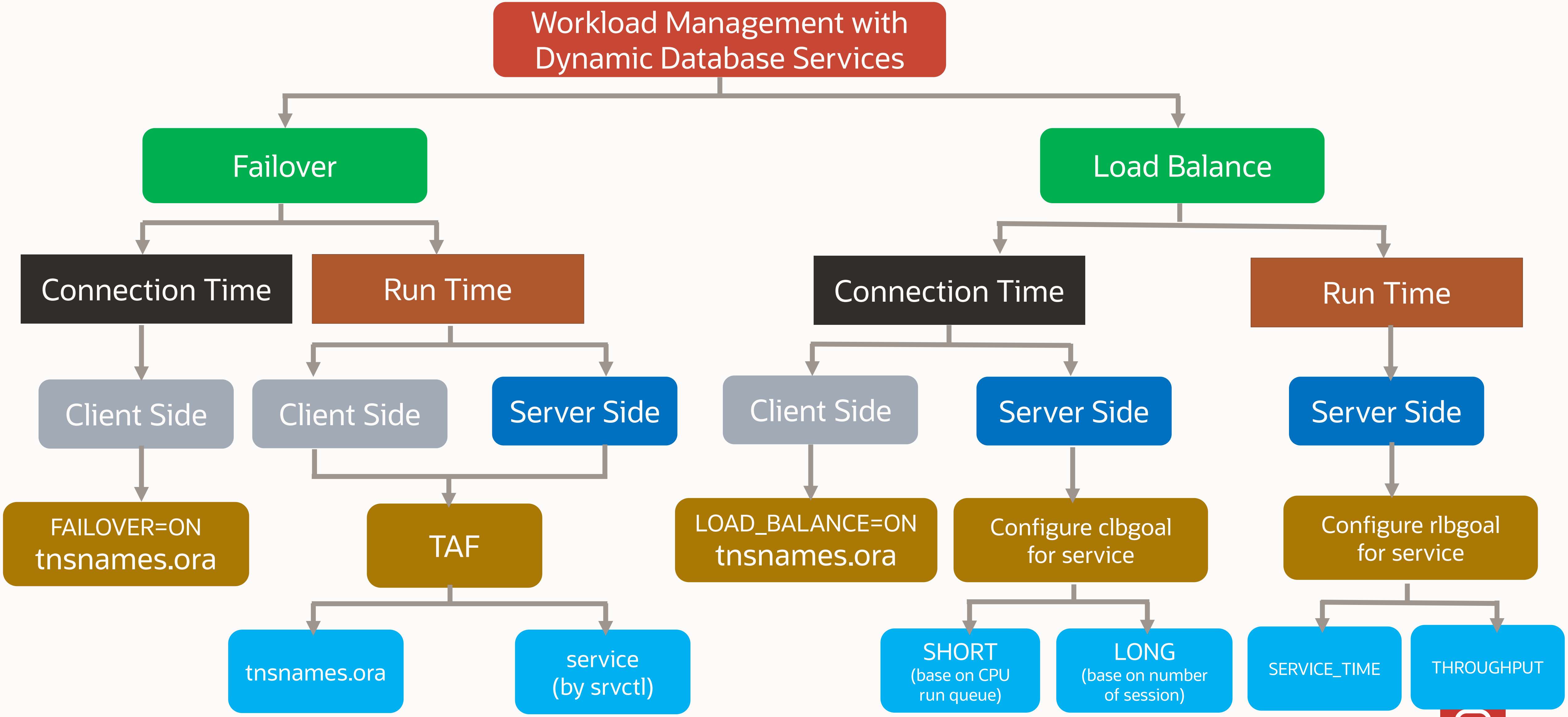


Figure 2: Data Guard/Active Data Guard Architecture

使用动态数据库服务进行工作负载管理



数据库Service的配置项

- ✓ CDB/PDB名
- ✓ 首选节点，备用节点
- ✓ 是否failback
- ✓ 负载均衡的方式
- ✓ 使用AC 或 TAC

- ✓ 使用事务卫士
- ✓ 回放的时间参数
- ✓ TAF配置
- ✓ FAN配置
- ✓ 排空的配置

```
$ srvctl add service -d <DATABASE NAME> -s <SERVICE NAME> [-pdb <PDB NAME>]
                         -preferred <INSTANCE NAME, …> -available <INSTANCE NAME, …>
  RAC HA             {   -notification true
                         -reset_state level1 [21C+]
                         -failover_restore AUTO
  TAC                {   -commit_outcome TRUE
                         -failovertype AUTO
                         -replay_init_time 600
  Workload               -drain_timeout 300
  Draining           {   -stopoption IMMEDIATE
                         -role PRIMARY
```

注意：不要使用默认数据库服务。不要使用默认的数据库服务来实现高可用性，因为此服务无法启用（enable）或禁用（ disable ），也无法在Oracle RAC上重新定位或切换到Oracle Data Guard。此服务是为Oracle Enterprise Manager Cloud Control （OEMCC）和DBA保留的(用于管理的目的)。

检查TAC服务配置信息：

```
$ srvctl config service  -db db19c -service TACS1
Service name: TACS1
Server pool:
Cardinality: 2
Service role: PRIMARY
Management policy: AUTOMATIC
DTP transaction: false
AQ HA notifications: true
Global: false
Commit Outcome: true
Failover type: AUTO
Failover method:
Failover retries: 30
Failover delay: 10
Failover restore: AUTO
Connection Load Balancing Goal: LONG
Runtime Load Balancing Goal: NONE
TAF policy specification: NONE
Edition:
Pluggable database name:
Hub service:
Maximum lag time: ANY
SQL Translation Profile:
Retention: 86400 seconds
Replay Initiation Time: 600 seconds
Drain timeout:
Stop option:
Session State Consistency: AUTO
GSM Flags: 0
Service is enabled
Preferred instances: db19c1,db19c2
Available instances:
CSS critical: no
Service uses Java: false
```

启用应用程序中的可变函数

✓ 可变函数（Mutable functions ）：是指每次执行时可能返回不同值的函数。

✓ 支持保留SYSDATE、SYSTIMESTAMP、LOCAL_TIMESTAMP、 SYS_GUID 、CURRENT_TIMESTAMP和sequence.NEXTVAL可变函数原始结果，如果不保留原始值，并且在重放时将不同的值返回给应用程序，则会拒绝重放。

✓ 针对SQL，应用程序连续性19c和更高版本自动启用可变函数的原值重放，因此不需要执行任何操作。

✓ 如果您使用PL/SQL的可变函数，或者您使用的是Oracle database 19c之前的数据库版本，那么DBA必须发出GRANT KEEP授权。当授予KEEP权限后，在重放时，会使用原始函数结果，例如：

对象的拥有者：

```
SQL> CREATE SEQUENCE.. [sequence object]  KEEP;
SQL> ALTER  SEQUENCE.. [sequence object] KEEP;
```

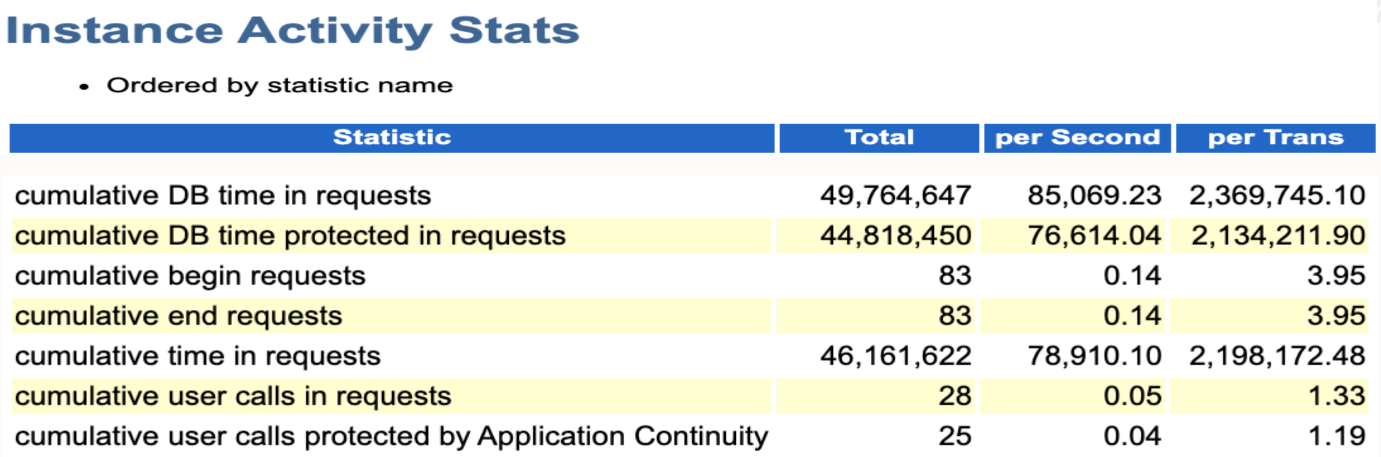
对其它用户授权：

```
SQL> GRANT KEEP DATE TIME  TO USER;
SQL> GRANT KEEP SYSGUID TO USER;
SQL> GRANT KEEP SEQUENCE  ON [sequence object] TO USER;
```

✓ 只有串行处理计划才支持保留SYS_GUID值。使用并行查询时，应用程序连续性无法恢复SYS_GUID的原始值。

了解使用TAC或AC时的保护级别

应用程序连续性从系统、会话和服务层面收集统计信息，使您能够监视保护级别。统计信息在V$SYSSTAT、V$SESSTAT中可用，并且在启用服务统计信息时，在V$SERVICE_STATS中可用。这些统计信息保存到自动工作负载存储库（AWR）中，并在AWR报告中可用。如下图所示:



Instance Activity Stats

- Ordered by statistic name

Statistic	Total	per Second	per Trans
cumulative DB time in requests	49,764,647	85,069.23	2,369,745.10
cumulative DB time protected in requests	44,818,450	76,614.04	2,134,211.90
cumulative begin requests	83	0.14	3.95
cumulative end requests	83	0.14	3.95
cumulative time in requests	46,161,622	78,910.10	2,198,172.48
cumulative user calls in requests	28	0.05	1.33
cumulative user calls protected by Application Continuity	25	0.04	1.19

如果Cumulative user calls in request = cumulative user calls protected相等，并且都不为0，说明他们正在发挥作用。可以使用以下方法计算受保护的用户调用的百分比：Percentage of Protected Calls = cumulative user calls protected / cumulative user calls in request * 100受保护的调用的百分比是否可能会小于100%。这个取决于一些因素，比如

- ✓ 您可能正在使用JDBC具体类(不支持废弃的oracle.sql 具体类)
- ✓ 副作用（Side Effects）被禁用
- ✓ 可能正在使用不可恢复的状态
- ✓ 或者应用程序可能会选择为某些请求禁用应用程序连续性

如果您的应用没有受到100%的保护，可以acchk来查找您的应用覆盖率低于100%的原因。你的管理层可以通过评估影响来决定是采纳报告中的建议，还是不做任何行动。

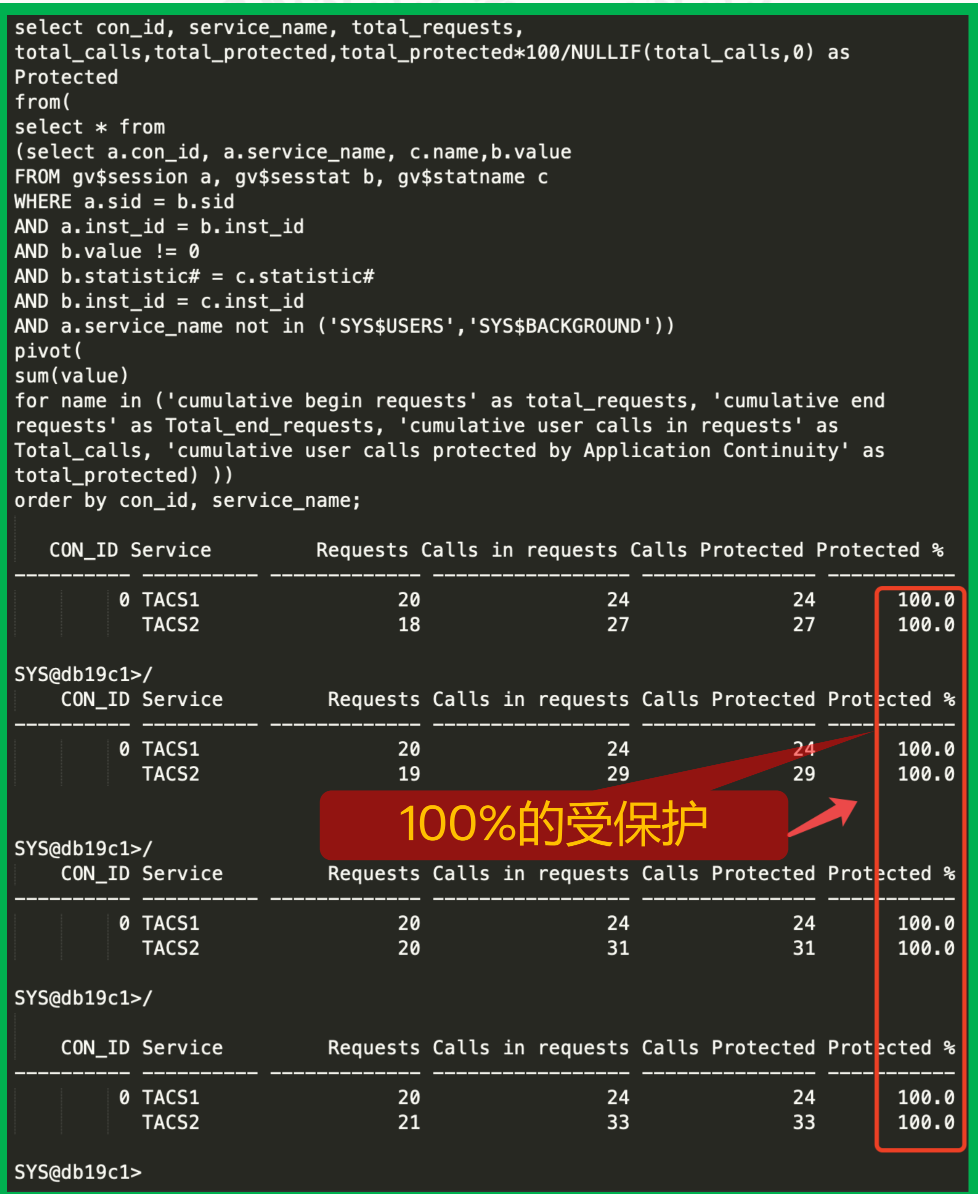
数据库中的统计信息

Statistic	19.3	19.12*	21c
cumulative begin requests	*	*	*
cumulative end requests	*	*	*
cumulative user calls in requests	*	*	*
cumulative user calls protected in requests by Application Continuity	*	*	*
cumulative DB time in requests		*	*
cumulative DB time protected in requests		*	*
successful replays by Application Continuity		*	*
rejected replays by Application Continuity		*	*

* 19c session statistics and ACCHK reports

使用SQL语句检查TAC的保护级别

```
set pagesize 60
set lines 120
col Service_name format a30 trunc heading "Service"
break on con_id skip1
col Total_requests format 999,999,9999 heading "Requests"
col Total_calls format 9,999,9999 heading "Calls in requests"
col Total_protected format 9,999,9999 heading "Calls Protected"
col Protected format 999.9 heading "Protected %"

select con_id, service_name, total_requests,
total_calls,total_protected,total_protected*100/NULLIF(total_calls,0) as
Protected
from(
select * from
(select a.con_id, a.service_name, c.name,b.value
FROM gv$session a, gv$sesstat b, gv$statname c
WHERE a.sid = b.sid
AND a.inst_id = b.inst_id
AND b.value != 0
AND b.statistic# = c.statistic#
AND b.inst_id = c.inst_id
AND a.service_name not in ('SYS$USERS','SYS$BACKGROUND'))
pivot(
sum(value)
for name in ('cumulative begin requests' as total_requests, 'cumulative end
requests' as Total_end_requests, 'cumulative user calls in requests' as
Total_calls, 'cumulative user calls protected by Application Continuity' as
total_protected) ))
order by con_id, service_name;
```



```
select con_id, service_name, total_requests,
total_calls,total_protected,total_protected*100/NULLIF(total_calls,0) as
Protected
from(
select * from
(select a.con_id, a.service_name, c.name,b.value
FROM gv$session a, gv$sesstat b, gv$statname c
WHERE a.sid = b.sid
AND a.inst_id = b.inst_id
AND b.value != 0
AND b.statistic# = c.statistic#
AND b.inst_id = c.inst_id
AND a.service_name not in ('SYS$USERS','SYS$BACKGROUND'))
pivot(
sum(value)
for name in ('cumulative begin requests' as total_requests, 'cumulative end
requests' as Total_end_requests, 'cumulative user calls in requests' as
Total_calls, 'cumulative user calls protected by Application Continuity' as
total_protected) ))
order by con_id, service_name;

   CON_ID Service         Requests Calls in requests Calls Protected Protected %
---------- ---------- ------------- ----------------- --------------- -----------
         0 TACS1                 20                24              24       100.0
           TACS2                 18                27              27       100.0

SYS@db19c1>/
    CON_ID Service         Requests Calls in requests Calls Protected Protected %
---------- ---------- ------------- ----------------- --------------- -----------
         0 TACS1                 20                24              24       100.0
           TACS2                 19                29              29       100.0


SYS@db19c1>/
    CON_ID Service         Requests Calls in requests Calls Protected Protected %
---------- ---------- ------------- ----------------- --------------- -----------
         0 TACS1                 20                24              24       100.0
           TACS2                 20                31              31       100.0

SYS@db19c1>/

    CON_ID Service         Requests Calls in requests Calls Protected Protected %
---------- ---------- ------------- ----------------- --------------- -----------
         0 TACS1                 20                24              24       100.0
           TACS2                 21                33              33       100.0

SYS@db19c1>
```

100%的受保护

使用ACCHK识别重放异常

启用acchk收集保护数据：

1）开启acchk设置：

```
SQL> EXECUTE DBMS_APP_CONT_ADMIN.ACCHK_SET(TRUE);
```

或：指定收集保护数据的时间，默认是600秒，也就是说开启acchk之后默认600秒会自动关闭保护数据的收集，测试要求在这个时间段内完成，可以指定更少的时间来禁用收集，如下：

```
SQL> EXECUTE DBMS_APP_CONT_ADMIN.ACCHK_SET(TRUE, 300);
```

2）连接到AC或TAC的数据库服务上进行测试：运行客户端程序；

3）测试完成后，关闭acchk设置：

```
SQL> EXECUTE DBMS_APP_CONT_ADMIN.ACCHK_SET(FALSE);
```

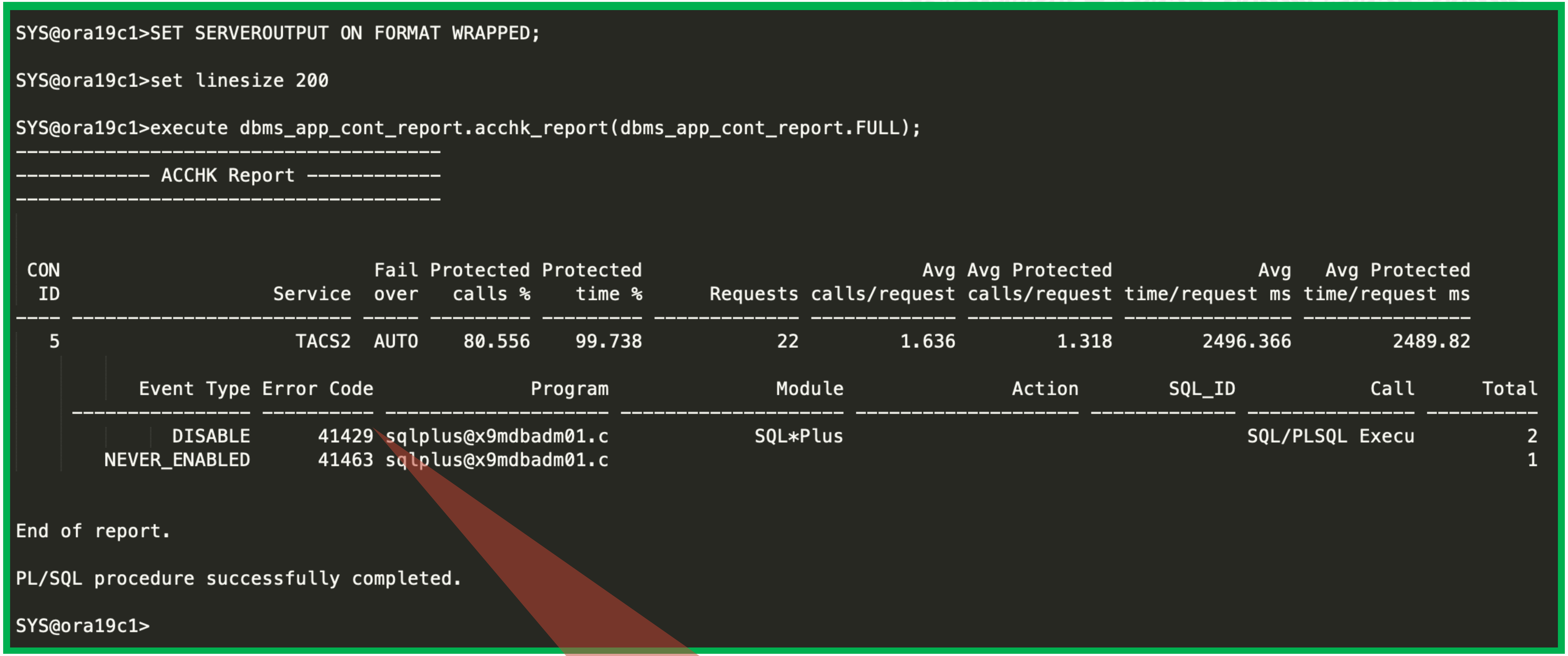
注意：数据采集将在600秒后自动禁用（默认值），或者可以在启用数据采集时设置计时器。
测试应在规定时间内完成。如果测试没有开始，或者在收集期内没有完成，
则要求重新启用ACCHK进行数据收集。

4）生成acchk报告：

```
SQL> SET SERVEROUTPUT ON FORMAT WRAPPED;
SQL> EXECUTE DBMS_APP_CONT_REPORT.ACCHK_REPORT(DBMS_APP_CONT_REPORT.FULL);
```

注意：报告级别为FULL、WARNING和SUMMARY，默认报告为SUMMARY。

Acchk报告示例



```
SYS@ora19c1>SET SERVEROUTPUT ON FORMAT WRAPPED;

SYS@ora19c1>set linesize 200

SYS@ora19c1>execute dbms_app_cont_report.acchk_report(dbms_app_cont_report.FULL);
--------------------------------------
------------ ACCHK Report ------------
--------------------------------------


 CON                            Fail Protected Protected                         Avg Avg Protected             Avg   Avg Protected
  ID                   Service  over   calls %    time %      Requests calls/request calls/request time/request ms time/request ms
---- ------------------------- ----- --------- --------- ------------- ------------- ------------- --------------- ---------------
   5                     TACS2  AUTO    80.556    99.738            22         1.636         1.318        2496.366         2489.82

           Event Type Error Code              Program               Module               Action       SQL_ID            Call      Total
     ---------------- ---------- -------------------- -------------------- -------------------- ------------- --------------- ----------
              DISABLE      41429 sqlplus@x9mdbadm01.c             SQL*Plus                                    SQL/PLSQL Execu          2
        NEVER_ENABLED      41463 sqlplus@x9mdbadm01.c                                                                                  1


End of report.

PL/SQL procedure successfully completed.

SYS@ora19c1>
```

Disable Reason
ORA-41429: side effect detected

目录

1. TAC简介

2. TAC应用

3. TAC限制

4. TAC演示

重放的限制和注意事项（禁用重放）-1

- ➢ 对重放Side Effects的控制

副作用(Side Effects)是外部操作，比如发送邮件、传输文件等。TAC会检测到副作用，并且不会重放它们。如果您希望重放副作用这些，那么可以使用AC，AC允许这种额外的灵活性。

副作用示例：

- ✓ DBMS_ALERT calls (email or other notifications)
- ✓ DBMS_FILE_TRANSFER calls (copying files)
- ✓ DBMS_PIPE and RPC calls (to external sources)
- ✓ UTL_FILE calls (writing text files)
- ✓ UTL_HTTP calls (making HTTP callouts)
- ✓ UTL_MAIL calls (sending email)
- ✓ UTL_SMTP calls (sending SMTP messages)
- ✓ UTL_TCP calls (sending TCP messages)
- ✓ UTL_URL calls (accessing URLs)

- ➢ 对重放DBLINK的控制

DBLINK上的重放被限制，如果您希望重放DBLINK上操作，那么可以使用AC，AC允许这种额外的灵活性。

免费咨询热线：400-699-8888

重放的限制和注意事项（禁用重放）-2

- ➢ Drivers:
  - ✓ JDBC OCI Driver (Type 2), ODP.Net Managed, ODBC,OLE DB,OCCI,Pro* precompilers
- ➢ Java Concrete Classes (oracle.sql deprecated):
  - ✓ OPAQUE, ANYDATA, STRUCT
- ➢ Two-phase commit XA
- ➢ Replay on a different database or after data loss:
  - ✓ GoldenGate、Logical Standby、3rd Party Replicalon
  - ✓ one flashed back, recovered incompletely by media recovery, or opened by Oracle Data Guard to an earlier point in time
- ➢ Database-resident Connection Pool (DRCP)
- ➢ ISOLATION_LEVEL=SERIALIZABLE
- ➢ Default database service or default PDB service:
  - ✓ must create own service
- ➢ Transaction no longer than 10min (Parameter：REPLAY_INIT_TIMEOUT)
- ➢ Do not use Easy Connect like username/password@host:port/service_name
  - ✓ 19c Easy Connect syntax (primary-vip,secondary-vip:1521/"sales.example.com?connect_timeout=90&transport_connect_timeout=3&retry_count=30&retry_delay=3")

重放的限制和注意事项（禁用重放）-3

Normal Runtime

Replay is disabled per request after

- a successful commit
- a disabling user call (supported: basic SQL and LOB use)
- OCIRequestDisableReplay API
- Some ALTER SESSION operations
- Database links ADG to primary

Replay

- Error is not recoverable
- Reconnection failure
  - ✓ replay initiation timeout
  - ✓ max failover retries
- Last call committed in embedded (DDL, PL/SQL) or autocommit mode
- Validation detects different results
- Alter system/database statement

目录

1. TAC简介

2. TAC应用

3. TAC限制

4. TAC演示

TAC演示：

- ✓ 演示1：RAC环境正常关闭实例—DML操作无中断
- ✓ 演示2：RAC环境异常关闭实例—DML操作无中断
- ✓ 演示3：可变函数sysdate保留原始重放
- ✓ 演示4：可变序列保留原始值重放
- ✓ 演示5： DDL操作无中断
- ✓ 演示6：ADG切换DML操作无中断

环境描述（数据库版本19.13）：

- 创建服务：

```
[oracle@x9mdbadm01 ~]$ srvctl add service -db db19c -service TACS1 -preferred db19c1,db19c2 -failover_restore AUTO -commit_outcome TRUE -failovertype AUTO -replay_init_time 600 -retention 86400 -notification TRUE -drain_timeout 60 -stopoption IMMEDIATE
```

- 配置连接串：

```
[oracle@x9mdbadm01 ~]$ vi $ORACLE_HOME/network/admin/tnsnames.ora
tacs1 = (DESCRIPTION =
  (CONNECT_TIMEOUT=90)(RETRY_COUNT=50)(RETRY_DELAY=3)(TRANSPORT_CONNECT_TIMEOUT=3)
   (ADDRESS_LIST =
     (LOAD_BALANCE=on)
     (ADDRESS = (PROTOCOL = TCP)(HOST=x9m-scan1)(PORT=1521)))
   (CONNECT_DATA=(SERVICE_NAME = TACS1)))
```

- 创建用户及授权：

```
SQL> create user u1 identified by We1come$ default tablespace users;
SQL> grant resource,connect to u1;
SQL> alter user u1 quota unlimited on users;
SQL> grant execute on DBMS_APP_CONT to u1;
SQL> grant xxx to u1;  --授权需要权限
```

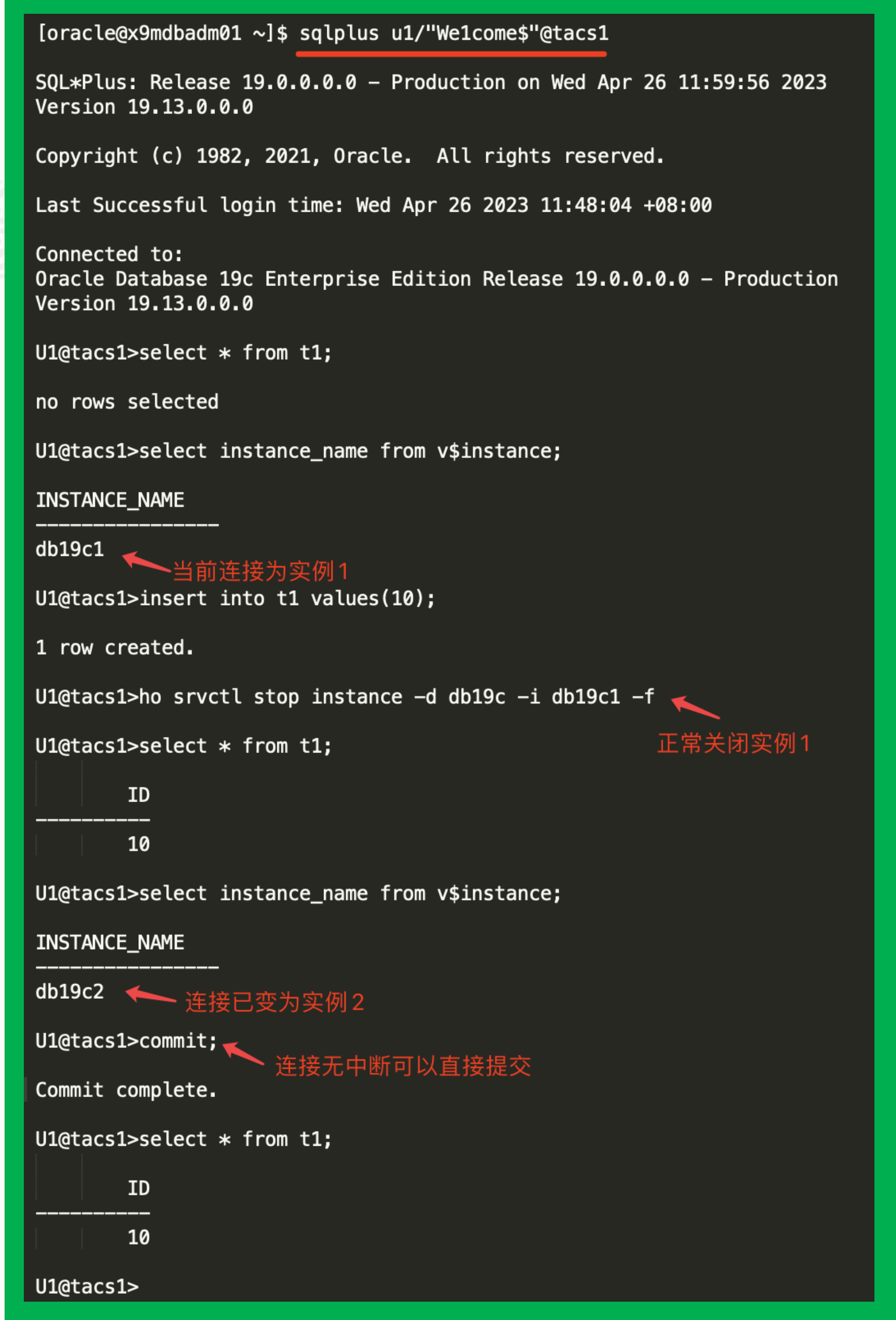
免费咨询热线：400-699-8888

演示1：RAC环境正常关闭实例—DML操作无中断

使用srvctl stop instance命令正常关闭实例数据库：

1.使用正确的连接串登录数据库；
2.查看连接的数据库实例（db19c1）；
3.向表中插入一条记录；
4.正常关闭当前登录的实例（db19c1）；
5.查看表中的记录；
6.查看当前连接的数据库实例（db19c2）；
7.提交；
8.验证插入记录；

测试结论：正常关闭实例，其上的DML操作无中断。



```
[oracle@x9mdbadm01 ~]$ sqlplus u1/"We1come$"@tacs1

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Apr 26 11:59:56 2023
Version 19.13.0.0.0

Copyright (c) 1982, 2021, Oracle.  All rights reserved.

Last Successful login time: Wed Apr 26 2023 11:48:04 +08:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.13.0.0.0

U1@tacs1>select * from t1;

no rows selected

U1@tacs1>select instance_name from v$instance;

INSTANCE_NAME
----------------
db19c1

U1@tacs1>insert into t1 values(10);

1 row created.

U1@tacs1>ho srvctl stop instance -d db19c -i db19c1 -f

U1@tacs1>select * from t1;

        ID
----------
        10

U1@tacs1>select instance_name from v$instance;

INSTANCE_NAME
----------------
db19c2

U1@tacs1>commit;

Commit complete.

U1@tacs1>select * from t1;

        ID
----------
        10

U1@tacs1>
```

当前连接为实例1

正常关闭实例1

连接已变为实例2

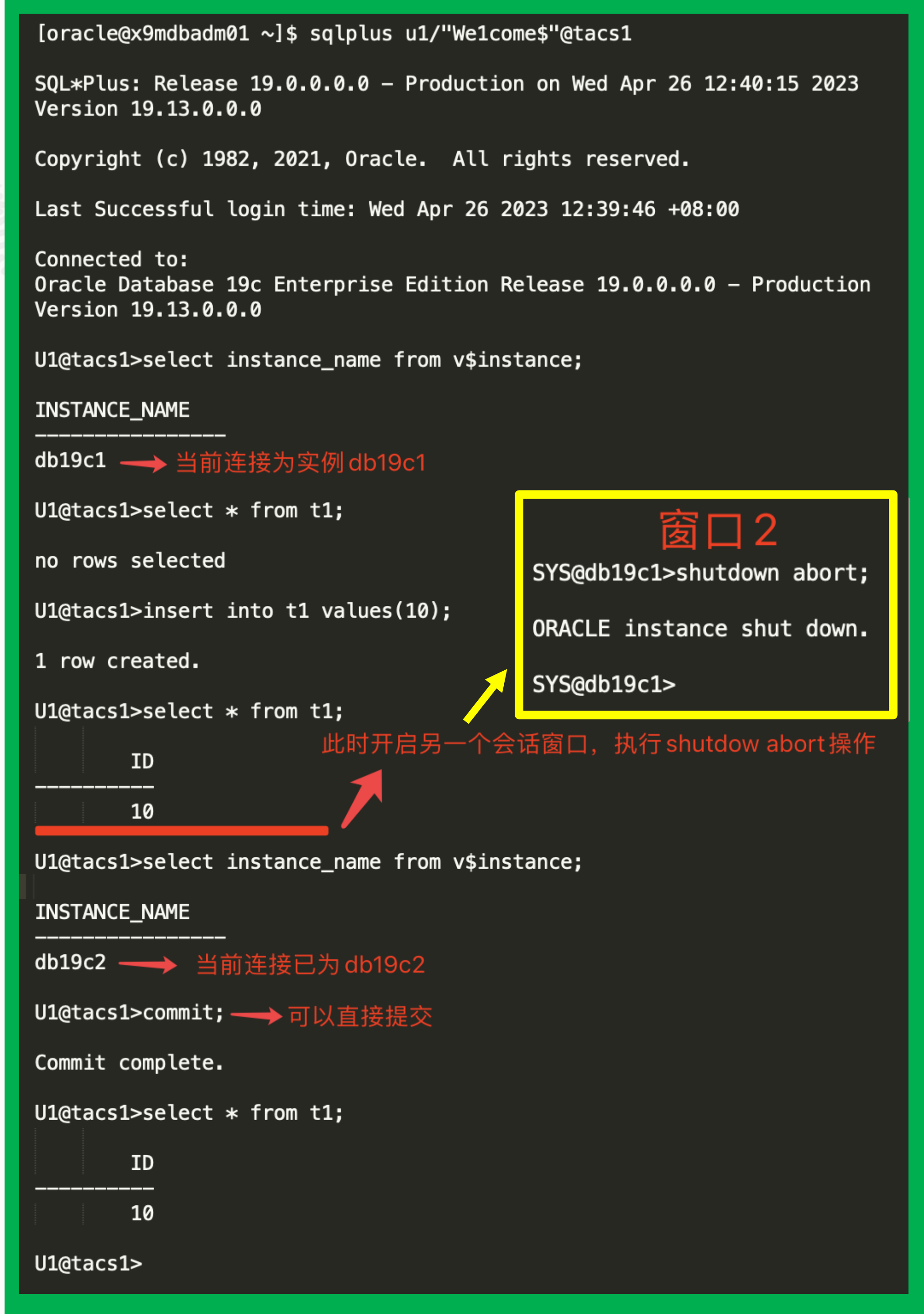
连接无中断可以直接提交

演示2：RAC环境异常关闭实例—DML操作无中断

使用shutdown abort命令异常关闭实例数据库：

1.使用正确的连接串登录数据库；
2.查看连接的数据库实例（db19c1）；
3.向表中插入一条记录；
4.在另一个窗口异常关闭当前登录的实例（db19c1）；
5.查看表中的记录；
6.查看当前连接的数据库实例（db19c2）；
7.提交；
8.验证插入记录；

测试结论：异常关闭实例，其上的DML操作无中断。



```
[oracle@x9mdbadm01 ~]$ sqlplus u1/"We1come$"@tacs1

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Apr 26 12:40:15 2023
Version 19.13.0.0.0

Copyright (c) 1982, 2021, Oracle.  All rights reserved.

Last Successful login time: Wed Apr 26 2023 12:39:46 +08:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.13.0.0.0

U1@tacs1>select instance_name from v$instance;

INSTANCE_NAME
----------------
db19c1

U1@tacs1>select * from t1;

no rows selected

U1@tacs1>insert into t1 values(10);

1 row created.

U1@tacs1>select * from t1;

        ID
----------
        10

U1@tacs1>select instance_name from v$instance;

INSTANCE_NAME
----------------
db19c2

U1@tacs1>commit;

Commit complete.

U1@tacs1>select * from t1;

        ID
----------
        10

U1@tacs1>
```

当前连接为实例db19c1

窗口2

```
SYS@db19c1>shutdown abort;

ORACLE instance shut down.

SYS@db19c1>
```

此时开启另一个会话窗口，执行shutdow abort操作

当前连接已为db19c2

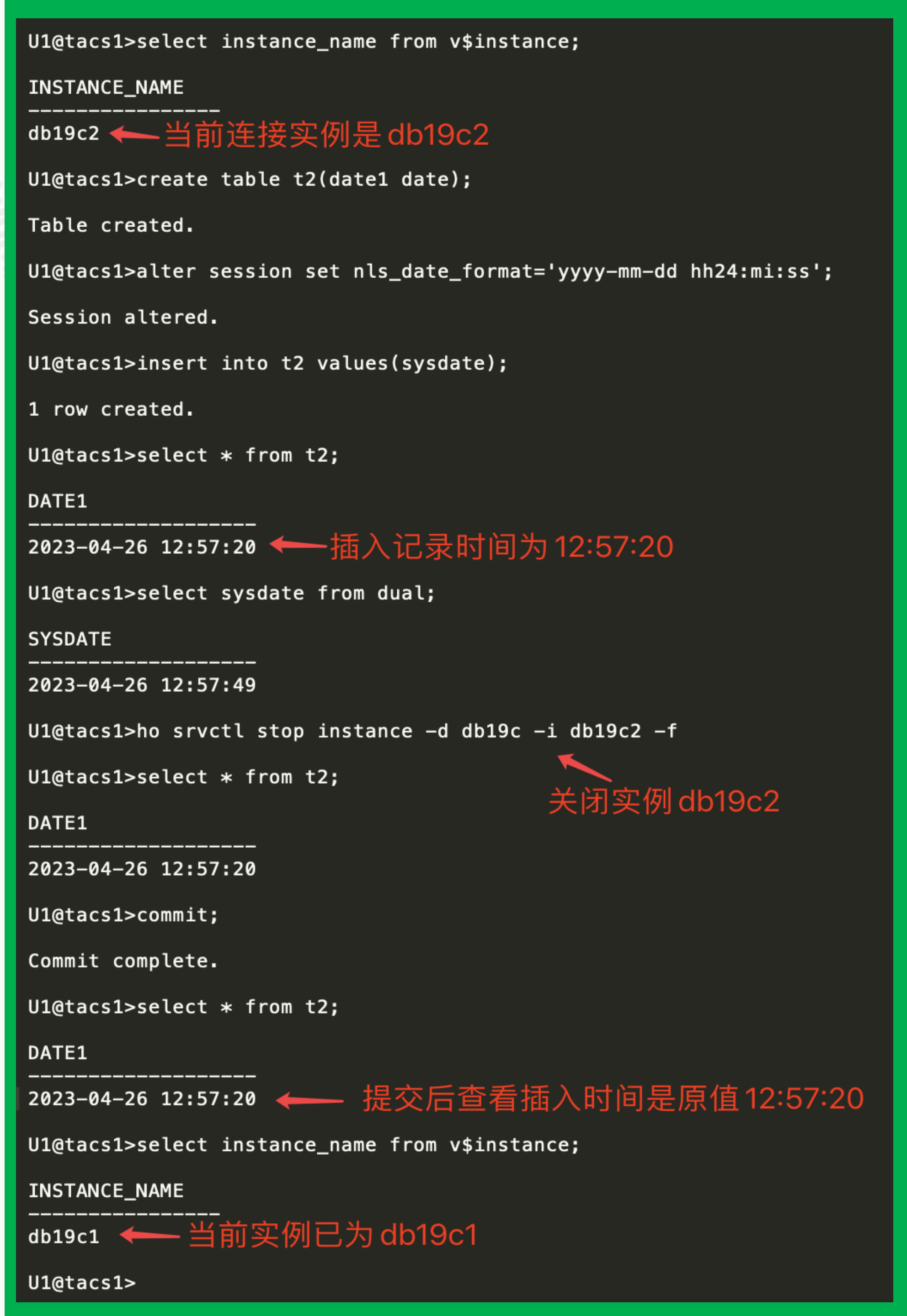
可以直接提交

演示3：可变函数sysdate保留原始重放

插入带sysdate的记录，关闭当前实例，检查插入值：

1.使用正确的连接串登录数据库；
2.查看连接的数据库实例（db19c2）；
3.向表中插入一条带sysdate的记录；
4.正常当前登录的关闭实例（db19c2）；
5.查看表中的记录；
6.提交；
7.验证插入记录；
8.查看当前连接的数据库实例（db19c1）；

测试结论：正常关闭实例，其上的DML操作无中断，插入的时间为中断前的原值。



```
U1@tacs1>select instance_name from v$instance;

INSTANCE_NAME
----------------
db19c2

U1@tacs1>create table t2(date1 date);

Table created.

U1@tacs1>alter session set nls_date_format='yyyy-mm-dd hh24:mi:ss';

Session altered.

U1@tacs1>insert into t2 values(sysdate);

1 row created.

U1@tacs1>select * from t2;

DATE1
-------------------
2023-04-26 12:57:20

U1@tacs1>select sysdate from dual;

SYSDATE
-------------------
2023-04-26 12:57:49

U1@tacs1>ho srvctl stop instance -d db19c -i db19c2 -f

U1@tacs1>select * from t2;

DATE1
-------------------
2023-04-26 12:57:20

U1@tacs1>commit;

Commit complete.

U1@tacs1>select * from t2;

DATE1
-------------------
2023-04-26 12:57:20

U1@tacs1>select instance_name from v$instance;

INSTANCE_NAME
----------------
db19c1

U1@tacs1>
```

当前连接实例是 db19c2

插入记录时间为 12:57:20

关闭实例 db19c2

提交后查看插入时间是原值 12:57:20

当前实例已为 db19c1

演示4：可变序列保留原始值重放

插入带序列值的记录，关闭当前实例，检查插入值：

SQL>create sequence seq1 start with 1 increment by 1 nomaxvalue nocycle cache 100;

1.使用正确的连接串登录数据库；

2.查看连接的数据库实例（db19c2）；

3.向表中插入一条带序列值的记录；

4.正常当前登录的关闭实例（db19c2）；

5.查看表中的记录；

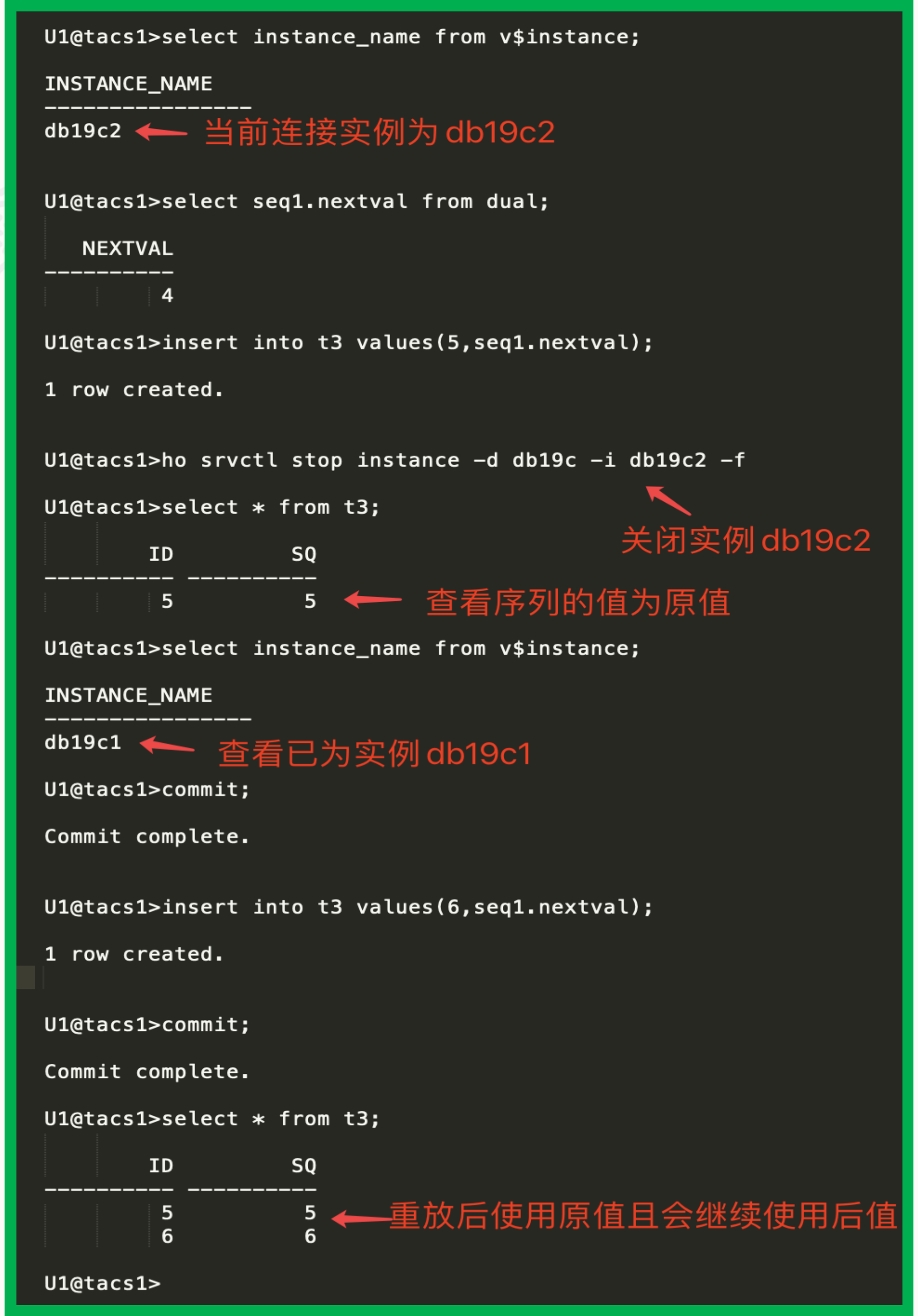
6.查看当前连接的数据库实例（db19c1）；

7.提交；

8.再次插入一条带序列值的记录；

9.验证插入记录；

测试结论：正常关闭实例，其上的DML操作无中断，插入的序列为中断前的原值，且会继续原值后的值。



```
U1@tacs1>select instance_name from v$instance;

INSTANCE_NAME
----------------
db19c2

U1@tacs1>select seq1.nextval from dual;

   NEXTVAL
----------
         4

U1@tacs1>insert into t3 values(5,seq1.nextval);

1 row created.

U1@tacs1>ho srvctl stop instance -d db19c -i db19c2 -f

U1@tacs1>select * from t3;

        ID         SQ
---------- ----------
         5          5

U1@tacs1>select instance_name from v$instance;

INSTANCE_NAME
----------------
db19c1

U1@tacs1>commit;

Commit complete.

U1@tacs1>insert into t3 values(6,seq1.nextval);

1 row created.

U1@tacs1>commit;

Commit complete.

U1@tacs1>select * from t3;

        ID         SQ
---------- ----------
         5          5
         6          6

U1@tacs1>
```

当前连接实例为 db19c2

关闭实例 db19c2

查看序列的值为原值

查看已为实例 db19c1

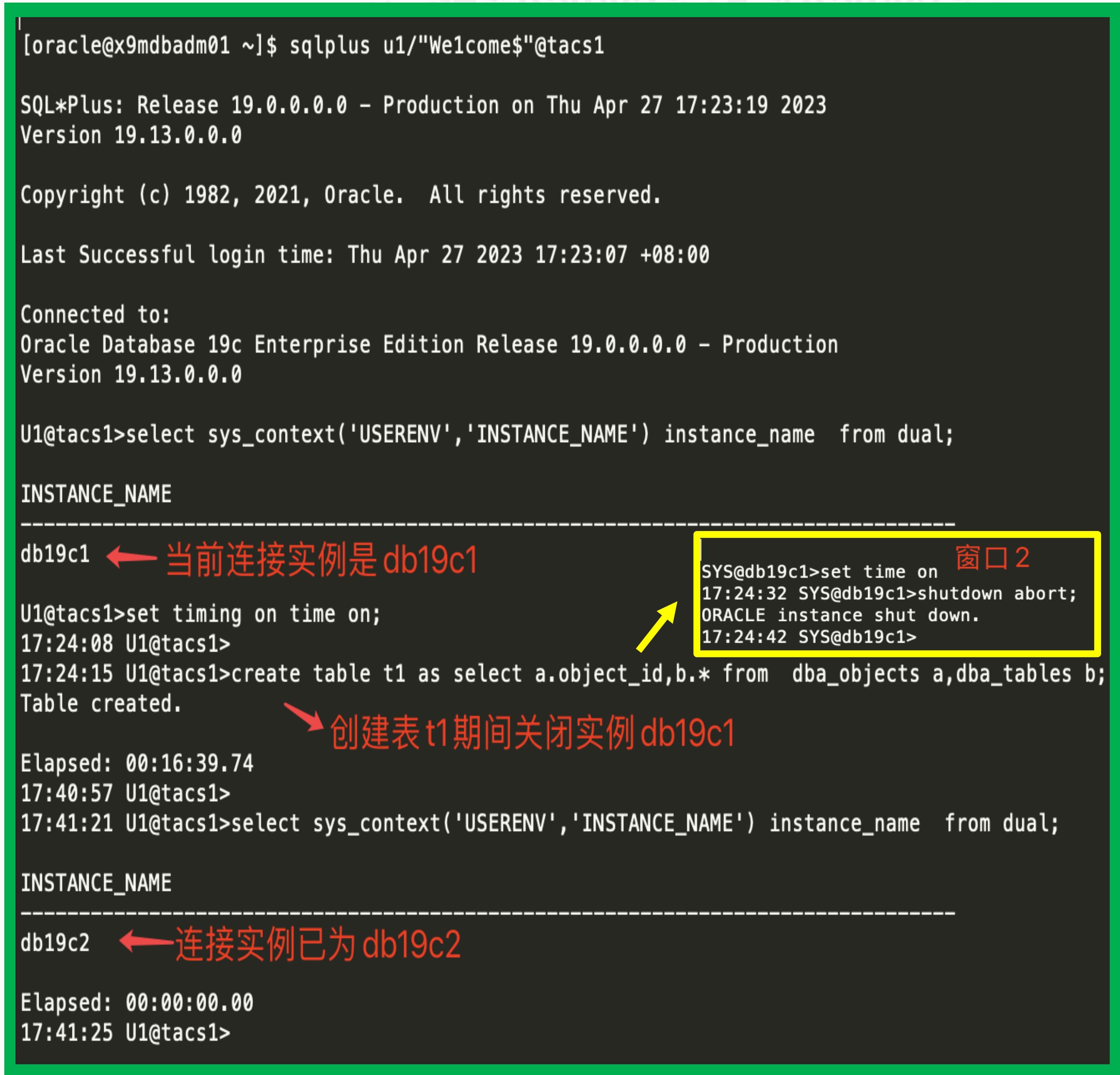
重放后使用原值且会继续使用后值

演示5： DDL操作无中断

执行DDL操作期间，关闭当前实例：

1.使用正确的连接串登录数据库；
2.查看连接的数据库实例（db19c1）；
3.使用CTAS命令创建一张表；
4.异常关闭数据库实例（db19c1）；
5.查看CTAS命令的执行状态；
6.查看当前连接的数据库实例（db19c2）；

测试结论：DDL操作期间关闭当前实例，DDL操作无中断，只是感觉到创建有些延迟而已。



```
[oracle@x9mdbadm01 ~]$ sqlplus u1/"We1come$"@tacs1

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Apr 27 17:23:19 2023
Version 19.13.0.0.0

Copyright (c) 1982, 2021, Oracle.  All rights reserved.

Last Successful login time: Thu Apr 27 2023 17:23:07 +08:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.13.0.0.0

U1@tacs1>select sys_context('USERENV','INSTANCE_NAME') instance_name  from dual;

INSTANCE_NAME
--------------------------------------------------------------------------------
db19c1

U1@tacs1>set timing on time on;
17:24:08 U1@tacs1>
17:24:15 U1@tacs1>create table t1 as select a.object_id,b.* from  dba_objects a,dba_tables b;
Table created.

Elapsed: 00:16:39.74
17:40:57 U1@tacs1>
17:41:21 U1@tacs1>select sys_context('USERENV','INSTANCE_NAME') instance_name  from dual;

INSTANCE_NAME
--------------------------------------------------------------------------------
db19c2

Elapsed: 00:00:00.00
17:41:25 U1@tacs1>
```

```
SYS@db19c1>set time on
17:24:32 SYS@db19c1>shutdown abort;
ORACLE instance shut down.
17:24:42 SYS@db19c1>
```

演示6：ADG切换DML操作无中断(1)

- **创建服务：**

  创建主库服务：

```
   [oracle@x9mdbadm01 ~] $srvctl add service -db x9tstdb -service TACSDG -preferred x9tstdb1,x9tstdb2
-failover_restore AUTO -commit_outcome TRUE -failovertype AUTO -replay_init_time 1000 -retention
86400 -notification TRUE -role PRIMARY
```

  创建备库服务：

```
   [oracle@x8mdbadm01 ~] srvctl add service -db x9std -service TACSDG -preferred x9std1,x9std2 -
failover_restore AUTO -commit_outcome TRUE -failovertype AUTO -replay_init_time 1000 -retention
86400 -notification TRUE -role PRIMARY
```

- **配置连接串：**

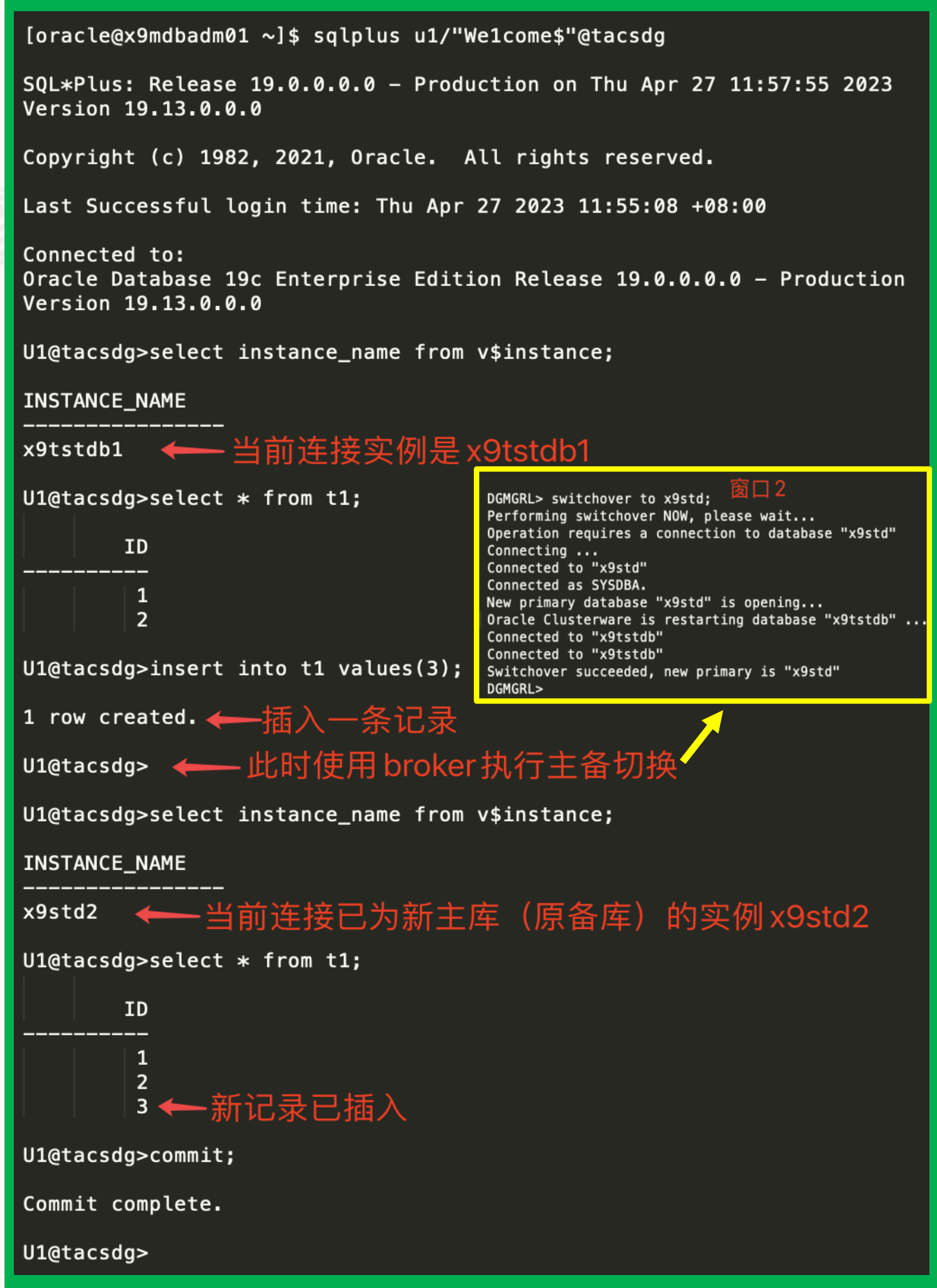
```
   [oracle@x9mdbadm01 ~]$ vi $ORACLE_HOME/network/admin/tnsnames.ora
   tacsdg = (DESCRIPTION =
     (CONNECT_TIMEOUT=
90)(RETRY_COUNT=50)(RETRY_DELAY=3)(TRANSPORT_CONNECT_TIMEOUT=3)
      (ADDRESS_LIST =
        (LOAD_BALANCE=on)
        (ADDRESS = (PROTOCOL = TCP)(HOST=x9m-scan1)(PORT=1521)))   ← 主库Scan-ip
      (ADDRESS_LIST =
        (LOAD_BALANCE=on)
        (ADDRESS = (PROTOCOL = TCP)(HOST=x8m-scan1)(PORT=1521)))   ← 备库Scan-ip
      (CONNECT_DATA=(SERVICE_NAME = TACSDG)))
```

演示6：ADG切换DML操作无中断(2)

执行主备切换，查看DML操作状态：

1.使用正确的连接串登录数据库；
2.查看连接的数据库实例（主库x9tstdb1）；
3.向表中插入一条记录；
4.使用broker执行主备切换；
5.查看当前连接的数据库实例（新主库x9std2）；
6.查看表中的记录；
7.提交（成功）；

测试结论：主备库切换，原主库上的DML操作无中断。



```
[oracle@x9mdbadm01 ~]$ sqlplus u1/"We1come$"@tacsdg

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Apr 27 11:57:55 2023
Version 19.13.0.0.0

Copyright (c) 1982, 2021, Oracle.  All rights reserved.

Last Successful login time: Thu Apr 27 2023 11:55:08 +08:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.13.0.0.0

U1@tacsdg>select instance_name from v$instance;

INSTANCE_NAME
----------------
x9tstdb1
```

当前连接实例是 x9tstdb1

```
U1@tacsdg>select * from t1;

        ID
----------
         1
         2

U1@tacsdg>insert into t1 values(3);

1 row created.
```

插入一条记录

```
U1@tacsdg>
```

此时使用 broker执行主备切换

窗口2

```
DGMGRL> switchover to x9std;
Performing switchover NOW, please wait...
Operation requires a connection to database "x9std"
Connecting ...
Connected to "x9std"
Connected as SYSDBA.
New primary database "x9std" is opening...
Oracle Clusterware is restarting database "x9tstdb" ...
Connected to "x9tstdb"
Connected to "x9tstdb"
Switchover succeeded, new primary is "x9std"
DGMGRL>
```

```
U1@tacsdg>select instance_name from v$instance;

INSTANCE_NAME
----------------
x9std2
```

当前连接已为新主库（原备库）的实例 x9std2

```
U1@tacsdg>select * from t1;

        ID
----------
         1
         2
         3
```

新记录已插入

```
U1@tacsdg>commit;

Commit complete.

U1@tacsdg>
```

附加材料：

Oracle Technology Network (OTN) Home page for Application Continuity
http://www.oracle.com/goto/ac

Application Continuity：
Continuous Availability, Application Continuity for the Oracle Database
(https://www.oracle.com/technetwork/database/options/clustering/applicationcontinuity/applicationcontinuityformaa-6348196.pdf)

Ensuring Application Continuity
(https://docs.oracle.com/en/database/oracle/oracle-database/21/racad/ensuring- application-continuity.html#GUID-C1EF6BDA-5F90-448F-A1E2-DC15AD5CFE75)

Application Continuity with Oracle Database12c Release 2
(http://www.oracle.com/technetwork/database/options/clustering/applicationcontinuity/overview/application-continuity-wp-12c-1966213.pdf)

Graceful Application Switchover in RAC with No Application Interruption
My Oracle Support (MOS) Note: Doc ID 1593712.1

Embedding UCP with JAVA Application Servers:
WLS UCP Datasource：
https://blogs.oracle.com/weblogicserver/wls-ucp-datasource

Design and Deploy WebSphere Applications for Planned, Unplanned Database Downtimes and Runtime Load Balancing with UCP
(http://www.oracle.com/technetwork/database/application-development/planned-unplanned-rlb-ucp-websphere-2409214.pdf)

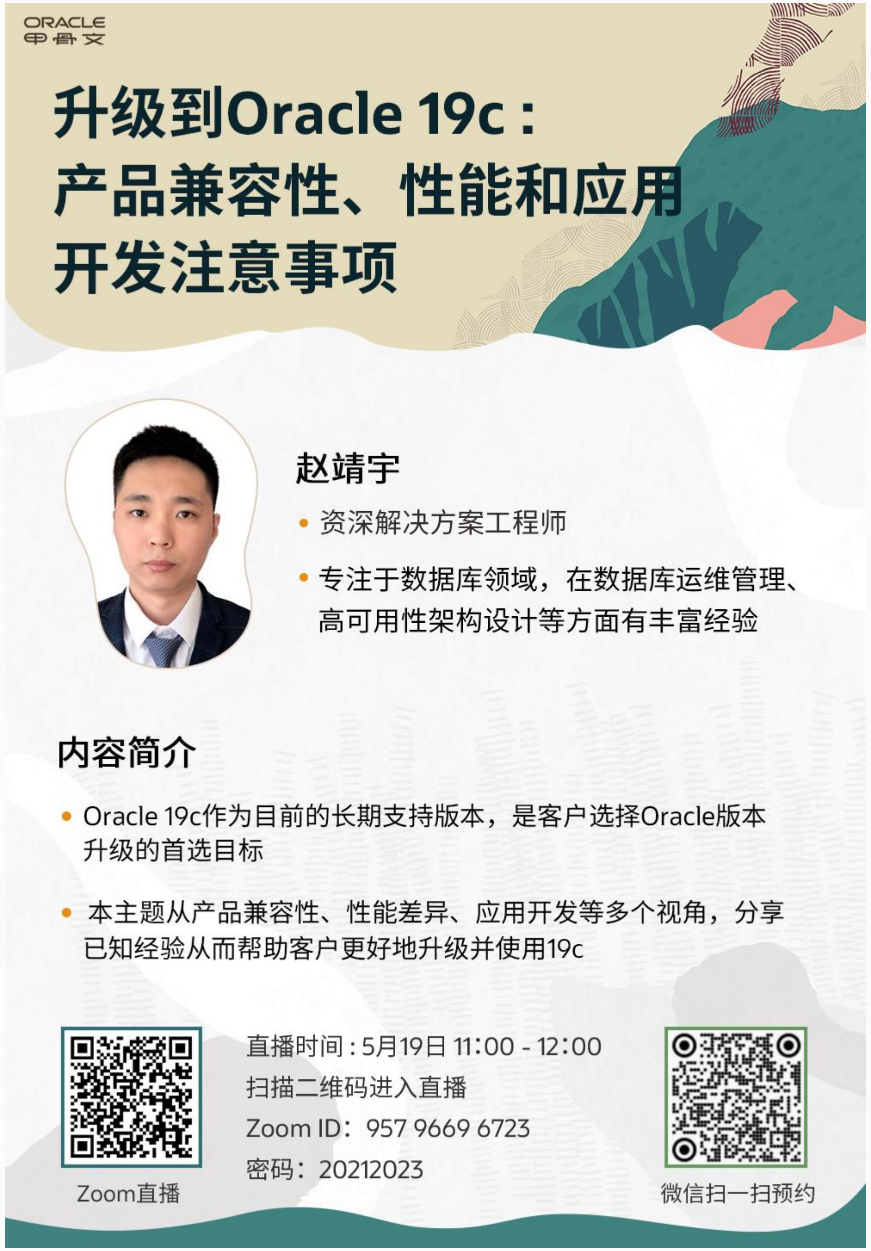
Reactive programming in microservices with MicroProfile on Open Liberty 19.0.0.4
(https://openliberty.io/blog/2019/04/26/reactive-microservices-microprofile-19004.html#oracle)

Design and deploy Tomcat Applications for Planned, Unplanned Database Downtimes and Runtime Load Balancing with UCP
(http://www.oracle.com/technetwork/database/application-development/planned-unplanned-rlb-ucp-tomcat-2265175.pdf).

Fast Application Notification：
http://www.oracle.com/technetwork/database/options/clustering/applicationcontinuity/learnmore/fastapplicationnotification12c-2538999.pdf

Q&A

谢 谢！



升级到Oracle 19c :
产品兼容性、性能和应用开发注意事项

赵靖宇

- 资深解决方案工程师
- 专注于数据库领域，在数据库运维管理、高可用性架构设计等方面有丰富经验

内容简介

- Oracle 19c作为目前的长期支持版本，是客户选择Oracle版本升级的首选目标
- 本主题从产品兼容性、性能差异、应用开发等多个视角，分享已知经验从而帮助客户更好地升级并使用19c

Zoom直播

直播时间 : 5月19日 11:00 - 12:00
扫描二维码进入直播
Zoom ID：957 9669 6723
密码：20212023

微信扫一扫预约

20-21

数据库和云讲座群

甲骨文云技术公众号

技术专家1V1深入交流

免费咨询热线：400-699-8888