

# アプリケーション18cの 透過的な高可用性

Real Application Clusters開発担当  
Carol Colrain 2018年5月



# プログラムのアジェンダ

- 1 Continuous Availabilityとは
- 2 計画メンテナンスの不可視化
- 3 透過的Application Continuity
- 4 最適な結果のための構成
- 5 お客様事例

# Continuous Availabilityの定義方法

## お客様によって定義はさまざま

Continuous Availabilityは絶対的な可用性ではありません。

データベース・レベルでの考えられる停止とメンテナンス・イベントはアプリケーションには表示されないため、アプリケーションはこれらのイベントを処理しながら、指定された応答時間の目標内でエラーなく動作し続けます。

### 重要な点:

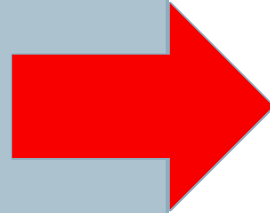
1. 計画メンテナンスと計画外停止(発生することがあります)が、アプリケーションに表示されない。
2. データの損失もデータの非一貫性も発生しない。
3. 大部分の作業(割合はお客様によって異なります)はリカバリ時間のSLA内に完了する。
4. やや遅れた実行として現れる場合がある。

**今日、多くのお客様はContinuous Availabilityを実現しています。**

# 高可用性とContinuous Availabilityの違い

## 高可用性

- 最小限の停止時間
- 処理中の作業の損失
- DBでのローリング・メンテナンス
- 予測可能な実行時パフォーマンス
- エラーが表示される場合あり
- 単一障害向けの設計
- 基本的なHAの基礎的要素



## Continuous Availability

- ユーザーの視点から停止時間を削除
- 処理中の作業を保持
- メンテナンスが表示されない
- 停止中の予測可能なパフォーマンス
- リカバリ不能な場合のみエラー
- 複数の同時障害向けの設計
- HAの上に構築

# データベース・リクエスト - 標準的なJDK9 - すべてのOracle 12c

標準的なJDK9 - すべてのOracle 12c

## 新しい概念

```
PoolDataSource pds = GetPoolDataSource();
```

```
Connection conn = pds.getConnection();
```

```
PreparedStatement pstmt = ...
```

...

SQL、PL/SQL、ローカル・コール、RPC

...

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

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

リクエストの  
開始

リクエスト本文は  
多くの場合COMMIT  
で終了

リクエストの  
終了

2 ▶ メンテナンス中はアプリケーションにエラーが表示されないようにする必要があります。

作業をドレインしましょう。



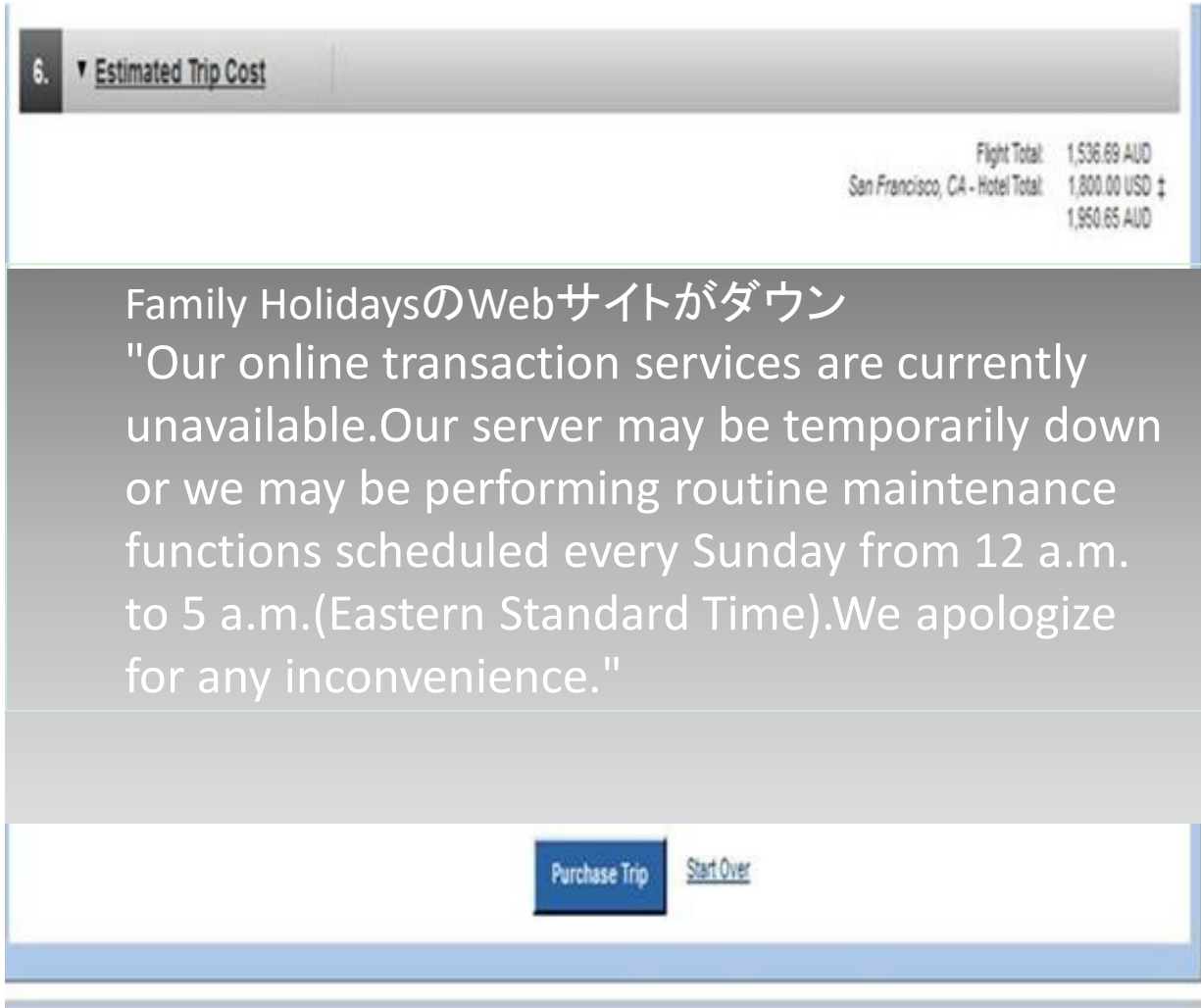
WHEN THINGS JUST SEEM  
TO BE GOING DOWN THE DRAIN...

# メンテナンス中はユーザーにエラーが表示されるべきではない

## 予防可能な状況

スケジュールされたデータベース・メンテナンス中にユーザーに停止時間を表示する理由はありません。

- サービスを使用できない
- アプリケーション所有者がメンテナンス期間に同意できない
- 長い時間実行されるジョブによってエラーが表示される
- DBAおよびエンジニアが勤務時間外に作業する
- アプリケーションとミドルウェアのコンポーネントを再起動する必要がある



6. Estimated Trip Cost

Flight Total:	1,536.69 AUD
San Francisco, CA - Hotel Total:	1,800.00 USD ±
	1,950.65 AUD

Family HolidaysのWebサイトがダウン  
"Our online transaction services are currently unavailable. Our server may be temporarily down or we may be performing routine maintenance functions scheduled every Sunday from 12 a.m. to 5 a.m. (Eastern Standard Time). We apologize for any inconvenience."

Purchase Trip Start Over

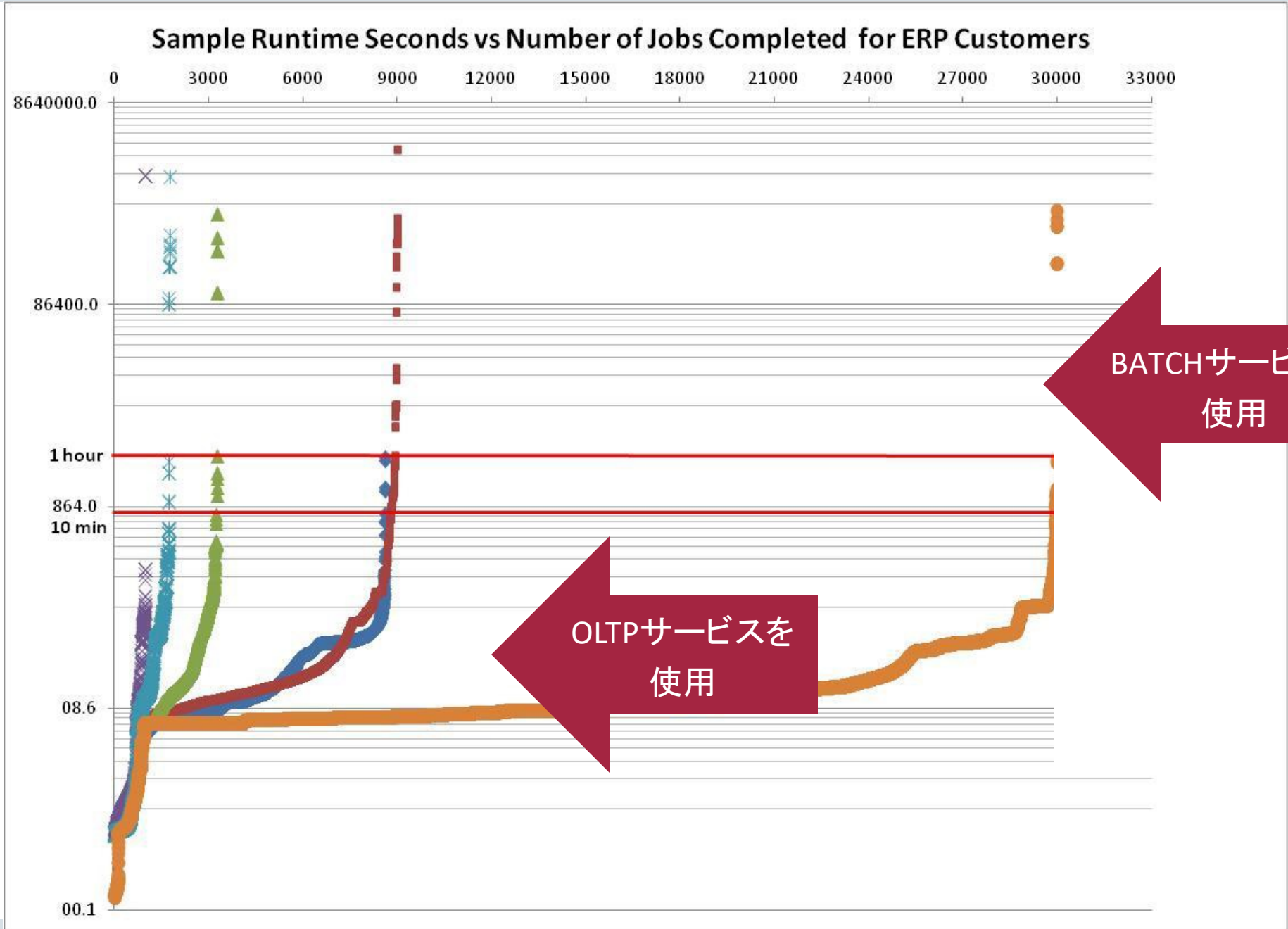
# すべてのアプリケーションに対する解決方法 (ディスカッション)

- ▶ アプリケーションにエラーが報告されることなく、別のインスタンス/データベースに作業を移動
- ▶ アプリケーションおよび中間層に対して透過的
- ▶ すべてのサーバー側のメンテナンスをサポート(パッチ、PSU、修理、変更、メジャー・リリース、切断/接続、移行、拡張、ハードウェアの交換)
- ▶ すべてのコマンドに対して同じものを1回構成



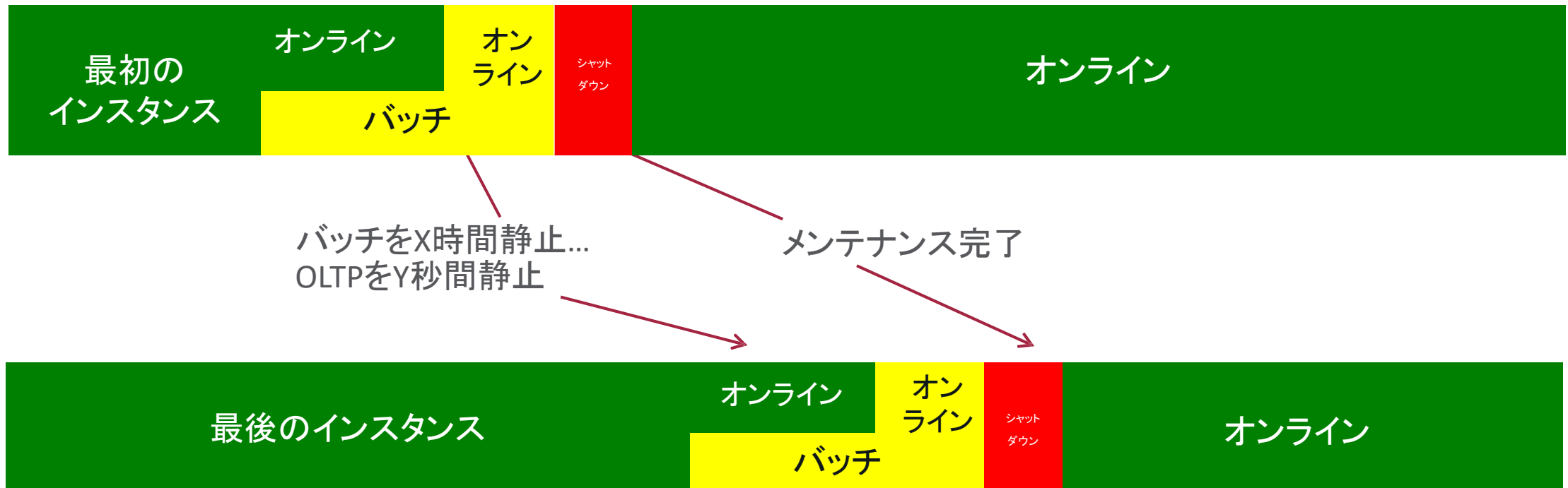
シンプルさを維持—  
バッチ、オンラインの順にドレイン

秒



# ドレインをずらして配置 – バッチ・サービス、オンライン・サービスの順

- 2つ以上のアプリケーション・サービスを使用 – オンラインおよびバッチ/バックエンド
- バッチのドレインを先に開始
- 数秒後にオンライン・ドレイン



# セッションをドレインすることは100%正しい



最初にサービスを移動

アプリケーションで認識されていない  
接続をドレイン

さらなるドレイン・ポイント

- 接続プール
- 接続テスト
- Webリクエスト
- トランザクション境界

# DBAの手順 – 11g以降

ドレイン時間の確保のためにサービスごとに繰り返す

- サービスの停止 (-forceなし)

```
srvctl stop service -db .. -instance .. -service .. (全体に対するサービスは含めない)
```

- またはサービスの再配置 (-forceなし)

```
srvctl relocate service -db .. -service .. -oldinst .. -newinst
```

```
srvctl relocate service -db .. -service .. -currentnode.. -targetnode
```

- セッションおよびトランザクションのドレインを待機
- 残ったセッションはトランザクションで停止

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

- お好きなツールを使用してインスタンスを停止 (無効化するオプション)

# Oracleプール – ドレインおよびリバランス

アプリケーションが  
使用しているもの

Oracle – WebLogic Active GridLink、ICC、UCP、ODP.NET  
管理対象および管理対象外、OCIセッション・プール、  
Tuxedo、(新規)CMANデータベース・トラフィック・マネー  
ジャ

UCPを使用するサード・パーティのアプリケーション・サーバー: IBM  
WebSphere、Apache Tomcat、NEC WebOTX、Red Hat JBoss、Spring

DBAの手順

```
srvctl [relocate | stop] service -drain_timeout
```

セッション・ドレイン

新しい作業はすぐにリダイレクト

段階的に

アクティブなセッションはプールに戻るときにリリース

FANを計画

ドレインと  
リバランス



# 他のJavaベースのアプリケーション・サーバーによるFAN-11g以降

## UCPの使用 - データソースのシンプルな置き換え

### 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

Apply OK Reset Cancel

### Additional Properties

Data sources

IBM WebSphere

Apache Tomcat

NEC WebOTX

RedHat JBoss

Spring Hibernate

OTNを参照してください。

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

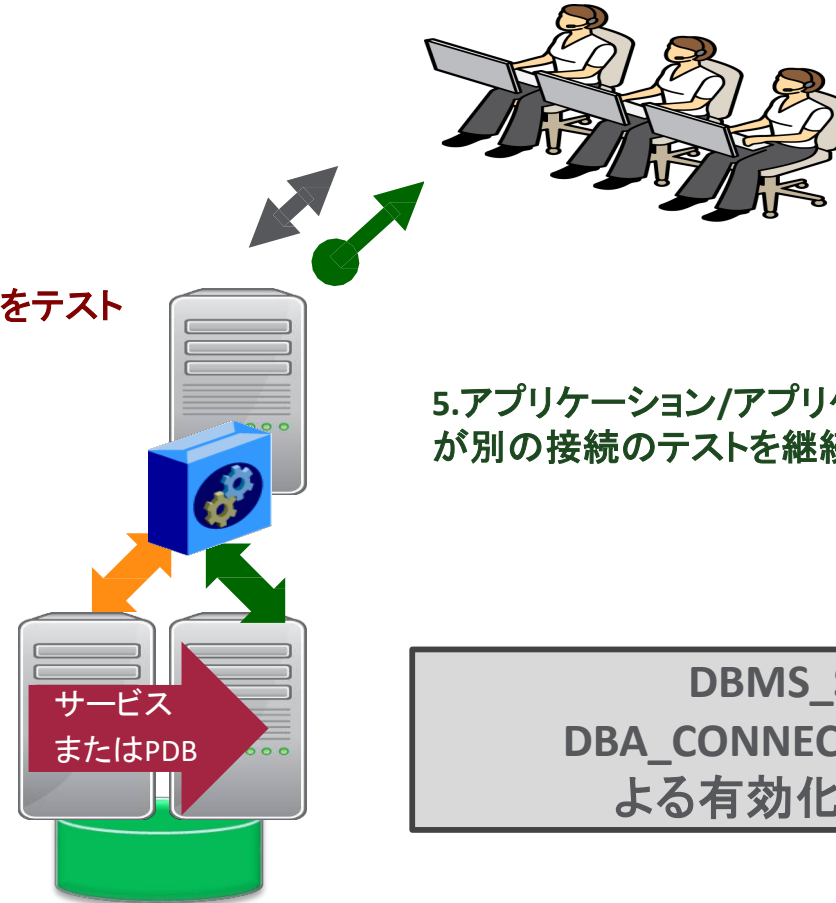
プール・データソース



# Oracleデータベースのドレイン

## 見えないところで - 接続テスト

1. サービスまたはPDBの停止または再配置
2. データベースがドレインするセッションをマーキング
3. アプリケーション/アプリケーション・サーバーが接続をテスト
4. データベースがルールを照合、"not good"を返して接続を終了



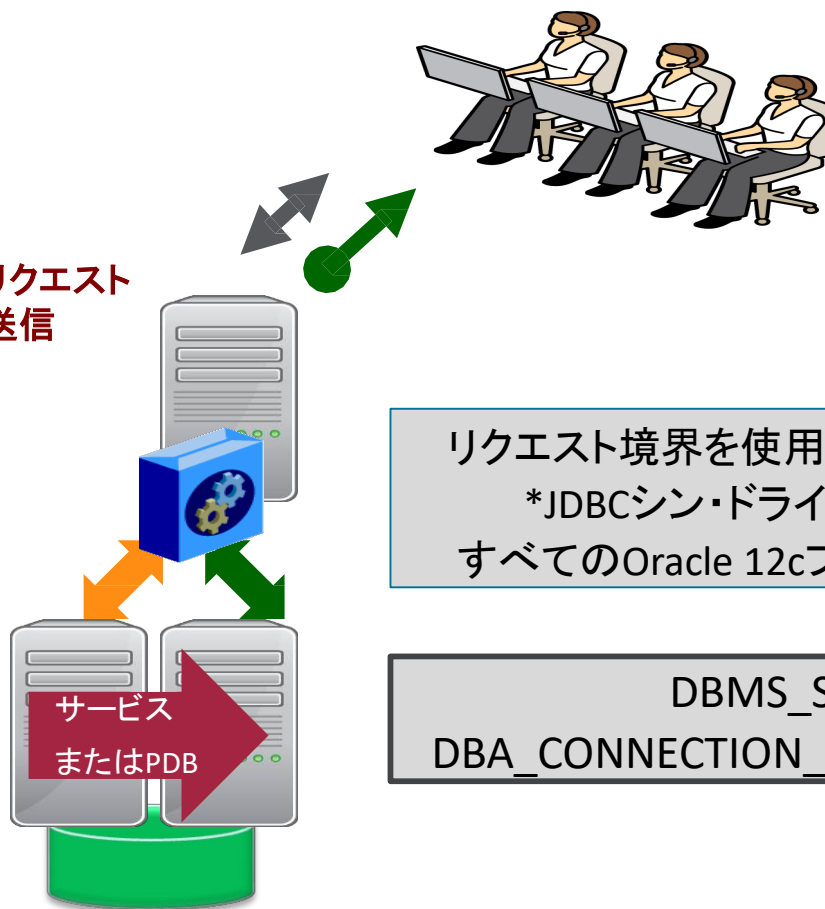
5. アプリケーション/アプリケーション・サーバーが別の接続のテストを継続

DBMS\_SESSION  
DBA\_CONNECTION\_TESTSによる有効化または追加

# Oracleデータベースのドレイン

## 見えないところで - Webリクエストの終了

3. アプリケーション/アプリケーション・サーバーがリクエスト境界を使用して、または使用せずに\*リクエストを送信
1. サービスまたはPDBの停止または再配置
2. データベースがドレインするセッションをマーキング
4. リクエストの終了時にデータベースがセッションを終了



リクエスト境界を使用したJDK9(一般提供)  
\*JDBCシン・ドライバが境界を検出  
すべてのOracle 12cプールおよびドライバ

DBMS\_SESSION  
DBA\_CONNECTION\_TESTSによる有効化



# 接続のテスト - データベースのドレイン

## SUPPORTS CURRENT AND EARLIER DRIVERS

NEW IN  
18<sup>c</sup>



- すべての標準SQLのテスト
- カスタムSQLのテスト
- Webリクエストの終了
- JDBC
  - isValid、pingDatabase、isUsable
- OCI
  - OCI Ping
  - OCI\_ATTR\_SERVER\_STATUS
- ODP.Net
  - connection.status

# アプリケーション・サーバーのドレイン

NEW IN  
18<sup>c</sup>

アプリケーション・サーバー	テスト名	DBへの接続テスト
Oracle WebLogic – 汎用データソースとマルチ・データ ソース	TestConnectionsOnReserve TestConnectionsOnCreate	isUsable  SQL – SELECT 1 FROM DUAL
Oracle WebLogic Active GridLink	embedded	isUsable
IBM WebSphere	PreTest Connections	SQL -SELECT 1 FROM DUAL
Red Hat JBoss	check-valid-connection-sql	SQL - SELECT COUNT(*) FROM DUAL
Apache TomCat	TestonBorrow TestonRelease	SQL - SELECT 1 FROM DUAL
ODP.NET管理対象外	Connection.status	OCI_ATTR_SERVER_STATUS

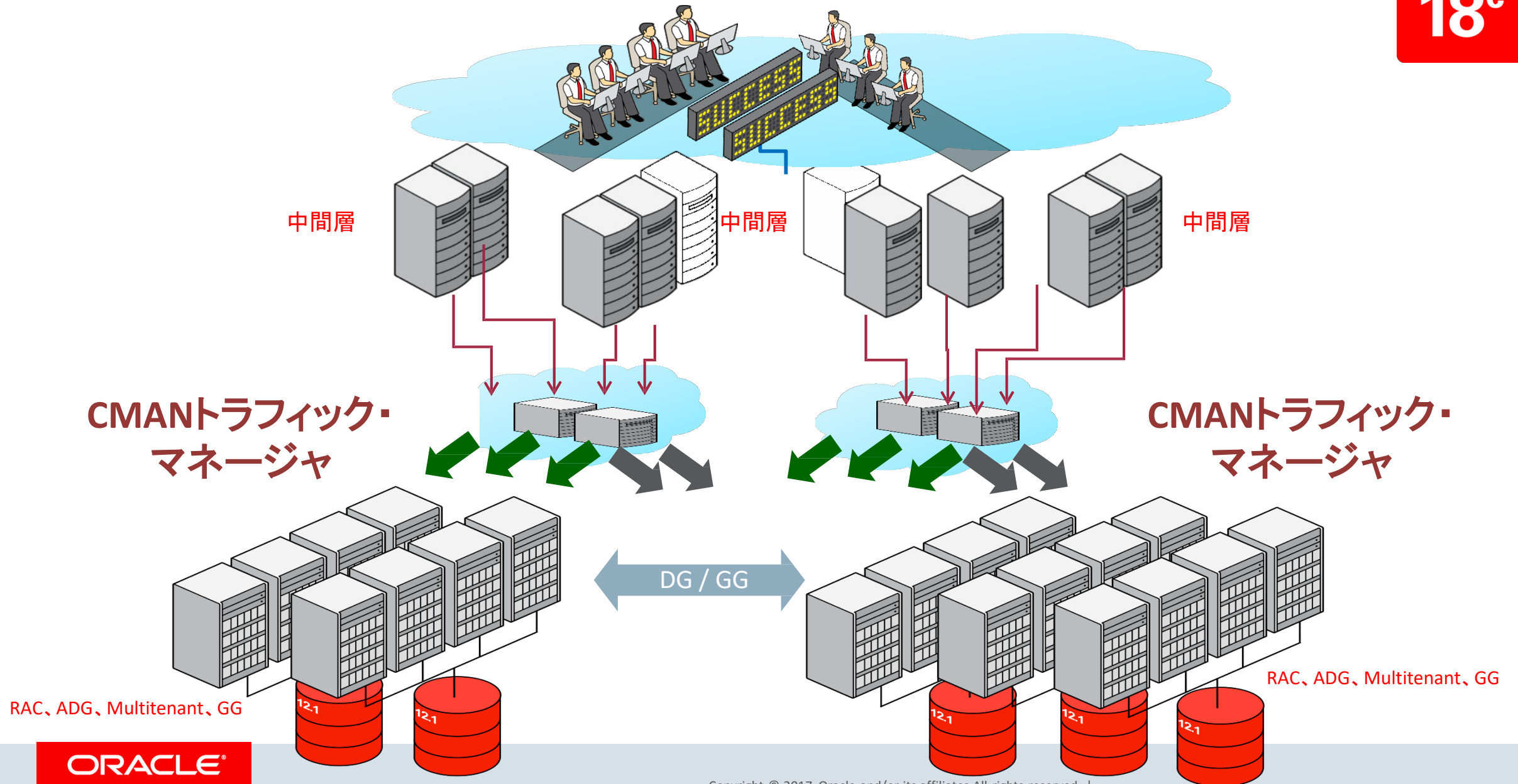
# アプリケーションのドレイン

NEW IN  
18<sup>c</sup>

アプリケーション	条件	DBへの接続テスト
E-Business Suite	Weblogicから接続が取得された	TestConnectionsOnReserveで "BEGIN NULL; END;"を指定
Fusion Applications	接続がWebLogicおよびC++プールに返されて、 チェックされた	TestConnectionsOnReserve isValidを指定 OCIPing OCI_ATTR_SERVER_STATUS
Siebel	接続がリクエストされた	OCI_ATTR_SERVER_STATUS
Peoplesoft	接続がリクエストされた	OCIPing
Customer	メタデータ表を使用したカスタム・プール 60秒ごとにステータスをチェック	OCI_ATTR_SERVER_STATUS

# CMANデータベース・トラフィック・マネージャ

NEW IN  
18<sup>c</sup>



# DBA作業の簡素化



# メンテナンス操作のための1つのコマンド – 12.2以降

ユーザー・エクスペリエンスを重視

- サービスのためのPDB、インスタンス、ノード、およびデータベースでの**グループ操作**
- 新しいサービス属性を1度に設定
- データベース、インスタンス、サービス、PDB、およびData Guardのスイッチオーバーに使用

**drain\_timeout** (秒)

**stopoption** (immediate、transactional)

# 1つのコマンド – 12.2以降

- データベース/ノード/PDBごとにすべてのサービスを再配置

```
srvctl relocate service -database -instance -drain_timeout.. -stopoption [immediate|transactional]
```

```
srvctl relocate service -node . -drain_timeout.. -stopoption [immediate|transactional]
```

```
srvctl stop service -pdb . -drain_timeout.. -stopoption [immediate|transactional]
```

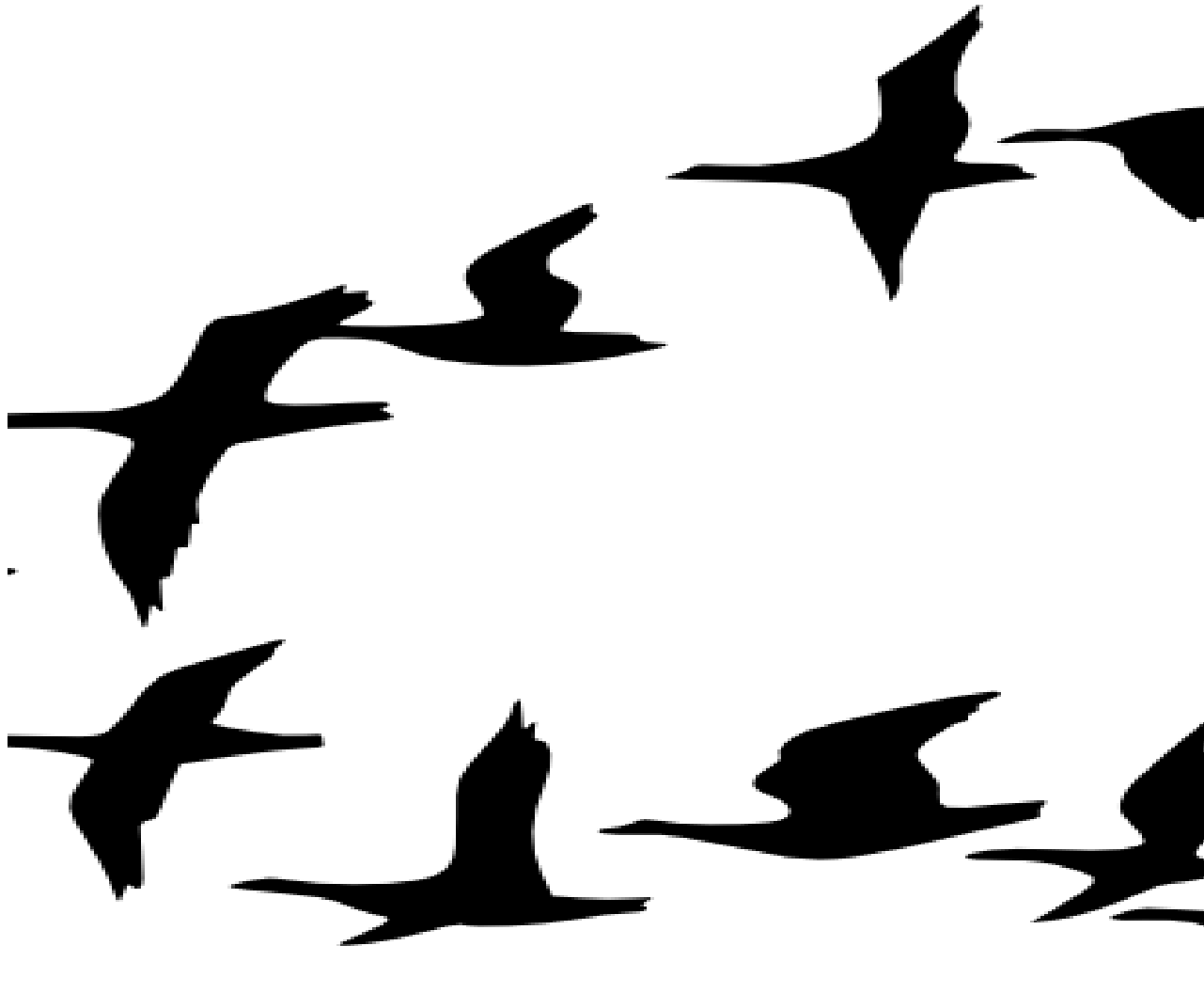
- 1つのノードですべてを開始/停止

```
srvctl stop service -node <node_name> -drain_timeout.. -stopoption [immediate|transactional]
```

```
srvctl stop database -node <node_name> -drain_timeout.. -stopoption [immediate|transactional]
```

- Data Guardのスイッチオーバー

```
Switchover to <db_resource_name> [wait [time]];
```



## 計画ドレインのデモ



ドレインしないと何が起こるか

停止が発生した場合どうなるか



# 処理中の作業

## 12c以前の状況

データベースの停止によって処理中の作業が失われ、ユーザーおよびアプリケーションは困惑

- アプリケーションおよび中間層の再起動
- ユーザーのフラストレーション
- 作業の取消し
- 重複する送信
- 計画された停止であってもエラーが発生
- 開発者の苦痛

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

6. Estimated Trip Cost

Flight Total	1,536.69 AUD
San Francisco, CA - Hotel Total	1,800.00 USD ‡
	1,950.65 AUD

Trip Total: 3,487.35 AUD ‡  
3,218.00 USD

‡ Please note that this total is based on available information. The estimated cost may not include taxes and fees.

- Remember to obtain an original invoice for all your expenses where required under the Global Travel Policy. The invoice should always include the name and address of your Oracle company. Failure to obtain a proper invoice may increase Oracle's costs by up to 25%.

**Please Note:**  
If you do not receive a confirmation after clicking Purchase Trip (EX. receive a **blank error message** or an error message that states "You already have an active session in the online booking site"), please call CWT before clicking purchase trip again.

Purchase Trip [Start Over](#)

# Application Continuity

## 処理中の作業が継続

- リカバリ可能なエラーの発生時に処理中の作業を再実行
- ハードウェア、ソフトウェア、ネットワーク、ストレージ、セッションのエラーとタイムアウトをマスク
- 12.1のJDBCシン、UCP、WebLogic Server、サード・パーティのJavaアプリケーション・サーバー
- 12.2のOCI、ODP.NET管理対象外、XA、Tuxedo\*、SQL\*Plus、Pro\*でのJDBCシン
- **18cは透過的(TAC)**

6. ▼ Estimated Trip Cost

Flight Total:	1,536.69 AUD
San Francisco, CA - Hotel Total:	1,800.00 USD ‡
	1,950.65 AUD

Trip Total: 3,487.35 AUD ‡  
3,218.00 USD

‡ Please note that this total is based on available information. The estimated cost may not include taxes and fees.

• Remember to obtain an original invoice for all your expenses where required under the Global Travel Policy. The invoice should always include the name and address of your Oracle company. Failure to obtain a proper invoice may increase Oracle's costs by up to 25%.

Your order number is 175634. You are protected by Application Continuity

[Purchase Trip](#) [Start Over](#)

# 透過的Application Continuity

– TACでの標準化

NEW IN  
18<sup>C</sup>



# アプリケーションの透過性

- アプリケーションに関する知識に依存しない
- アプリケーションが変更されても保証される未来
- 使用している場合はクライアント構成変更なし
  - リクエスト境界に依存しない
  - クライアント構成に依存しない
- 保険への支払いを好む人はいない
  - 少ない実行時リソース、迅速な再実行

DBAが主導権  
を握る

# 見えないところで

## 通常ของการ操作

- クライアントでリクエストをマーキング:  
明示的および検出済み
- サーバーで、再実行するコールを決定し、セッションの状態を追跡し、副次的作用を無効化
- 指令を受け、クライアントで元のコール、その入力、その他の検証データを保持

## フェイルオーバー フェーズ1:再接続

- 再実行が有効であることを確認
- タイムラインを検証
- 新しい接続を作成
- ターゲット・データベースが再実行に適していることを確認
- Transaction Guardを使用して、コミットされた結果を保証

## フェイルオーバー フェーズ2:再実行

- セッションの状態をリストアして検証
- 保持していたコールを再実行して関数の値をリストア
- 結果、クライアントの状態、メッセージがオリジナルと一致することを確認
- 成功時にアプリケーションに制御を返す

18cの新機能

# 透過的Application Continuity (TAC)

NEW IN  
18<sup>c</sup>

## TACでの容易な標準化

リクエスト境界

- 継続的に拡張

初期状態

- 自動的にリストアされて検証される

セッションの状態

- 追跡されて記録される

副次的作用

- 検出されて無効化される

mutating関数

- Grant KEEP

サポートされる古いコール

- JDBC具象クラス

統計情報

- 常に保護レベルを把握

# TAC - リクエスト境界の継続的な拡張

- リストア可能な状態が検出されたときにリクエスト境界が拡張
- 以前に無効化されたキャプチャの再有効化
- 小さいキャプチャ・セットが意味するのは高速リカバリ
- プールに戻るのが依然としてベスト・プラクティス
- 18cでは最初の境界が必要



Java 18c向け透過的アプリケーション・コンティニューイティ



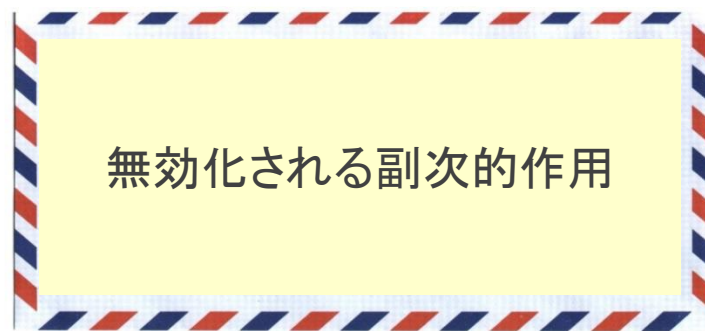
# 再実行前のセッションの初期状態

- 使用中のサービスでのFailover\_Restore (12.2以降)  
共通のセッション状態のリストア
- ログオン・トリガー、ラベル、TAF、またはJDBCコールバック
- 再実行前に複雑な状態のリストアを許可
- TACにより再実行前にオリジナルに一致するセッションを適用



# 再実行されない副次的作用

TAC – 次の有効化ポイントまで自動的に無効化



- 再実行しないリクエストがあるかどうかをTACが判断

例: DBMS\_LOGMNR  
DBMS\_SCHEDULER  
UTL\_HTTP  
UTL\_URL  
DBMS\_FILE  
DBMS\_FILE\_TRANSFER  
UTL\_SMTP  
UTL\_TCP  
UTL\_MAIL  
EXTPROC

- 自分で判断するには、ACを使用

# mutableの許可

再実行時に元の関数の結果を維持



所有シーケンスの場合：

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

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

他のユーザーに対する付与および取消し：

```
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]
```

# 常に保護レベルを把握

- AWR、システム、セッション、サービスの統計

- 1秒あたりのリクエスト完了件数
- リクエスト内のユーザー・コール
- 保護されたユーザー・コール

統計情報	合計	1秒あたり	トランザクションあたり
累積リクエスト	177,406	49.2	5.0
リクエスト内の累積ユーザー・コール	493,329	136.8	13.8
保護された累積ユーザー・コール	493,329	136.8	13.8

# 詳細な保護レポート(必要な場合)

## Application Continuity Summary

Outage Type	Status	Message
Coverage checks		<b>TotalRequest = 398</b> <b>PASS = 389</b> <b>WARNING = 0</b> <b>FAIL = 9</b>
	PASS	[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

# Application Continuityのパフォーマンス

Oracle SPEC-Jのパフォーマンス結果 – 12.1.0.2 DBおよび12.1.0.2 JDBC

	run#	DLR resp	Mfg resp	EjOPs	MT CPU	DB CPU
noreplay		27030.050/0.060/0.040	0.100/0.100	1117.889	27.94	27.77
replay		27020.050/0.060/0.050	0.100/0.100	1115.449	29.42	29.35

「アプリケーション・コンティニューイティ12102でのELAの低下なし

MT CPU = +5.3%

DB CPU = +5.7%

標準的なLinuxサーバーで実行(特殊なサーバーでない)」

適用されたベスト・プラクティス:

- ガベージ・コレクタの基本チューニング
- JDBCドライバでの文キャッシュ
- 高速表領域にTGを移動(EXAでは不要)

# 米国株式市場 – 本番前パフォーマンス検証

市場取引テスト・システム

50,000 TPS

アプリケーション・コンティニューイティによる97%を超えるスループット

アプリケーション・コンティニューイティに十分に投資したお客様は、  
優れた結果を得たことを認めている

適用されたベスト・プラクティス:

- ガベージ・コレクタの基本チューニング
- JDBCドライバでの文キャッシュ
- 高速表領域にTGを移動 (EXAでは不要)
- すべての推奨されるJDBCパッチ

2016年6月に一般提供

# データベースでの構成

## サービス属性

- FAILOVER\_TYPE = AUTOまたはTRANSACTION
- FAILOVER\_RESTORE = AUTOまたはLEVEL1(初期状態の場合)
- COMMIT\_OUTCOME = TRUE(Transaction Guardの場合)

次も確認:

- AQ\_HA\_NOTIFICATIONS = True(FAN OCIの場合)
- REPLAY\_INITIATION\_TIMEOUT = 300(再実行が取り消されるまでの秒数)
- Grant execute on DBMS\_APP\_CONT



4

基礎的要素が整備されているか



# 最適な結果のための構成

## 適切なサーバー・スタック

- RACまたはRAC One、Active Data Guard、GoldenGate、GDS

## Oracle Flex ASM

- Flex ASMまたはACFSのすべてのデータベース

## サービス

- 位置の透過性のためのサービス

## 継続的な接続

- 接続は継続的であるように見える

## 停止の通知

- 短期的中断に対するFAN

## 処理中の作業

- アプリケーション・コンティニューイティ

## リカバリ時間のSLA

- 適切なタイミングでのリカバリ

Start Now

# ローカルでのドレインおよびフェイルオーバー - サイト間スイッチオーバー

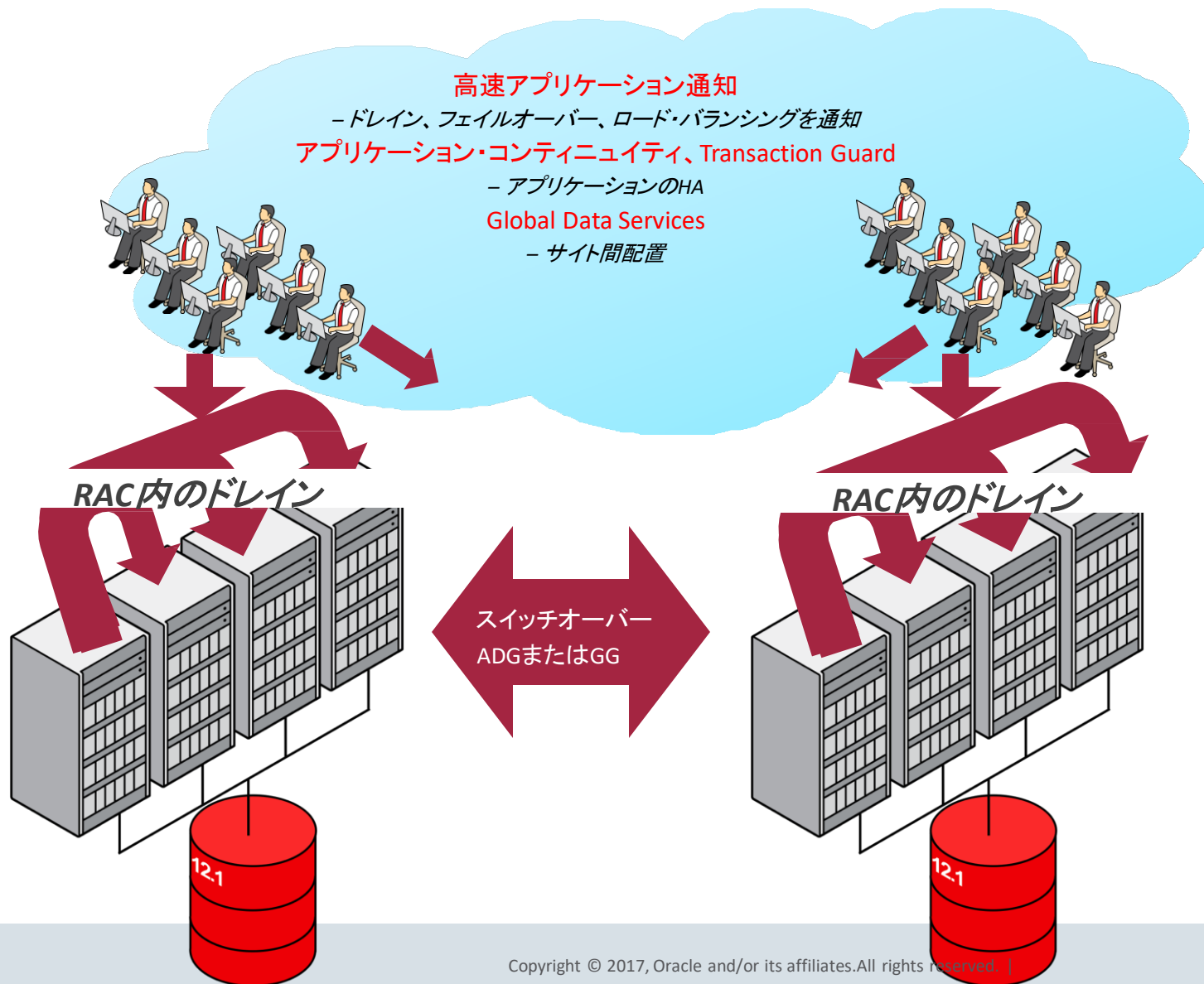
## 本番サイト

### Oracle RAC

- オンライン・ローリング・メンテナンス
- スケーラビリティ
- サーバーのHA

### Oracle RAC One

- オンライン・ローリング・メンテナンス
- サーバーのHA



## レプリカ

### Oracle Active Data Guard

- スケジュールされたスイッチオーバー
- データ保護、DR
- 問合せのオフロード

### Oracle Data Guard

- スケジュールされたスイッチオーバー
- データ保護、DR

### Oracle GoldenGate

- スケジュールされたスイッチオーバー
- アクティブ/アクティブ・レプリケーション
- 異種

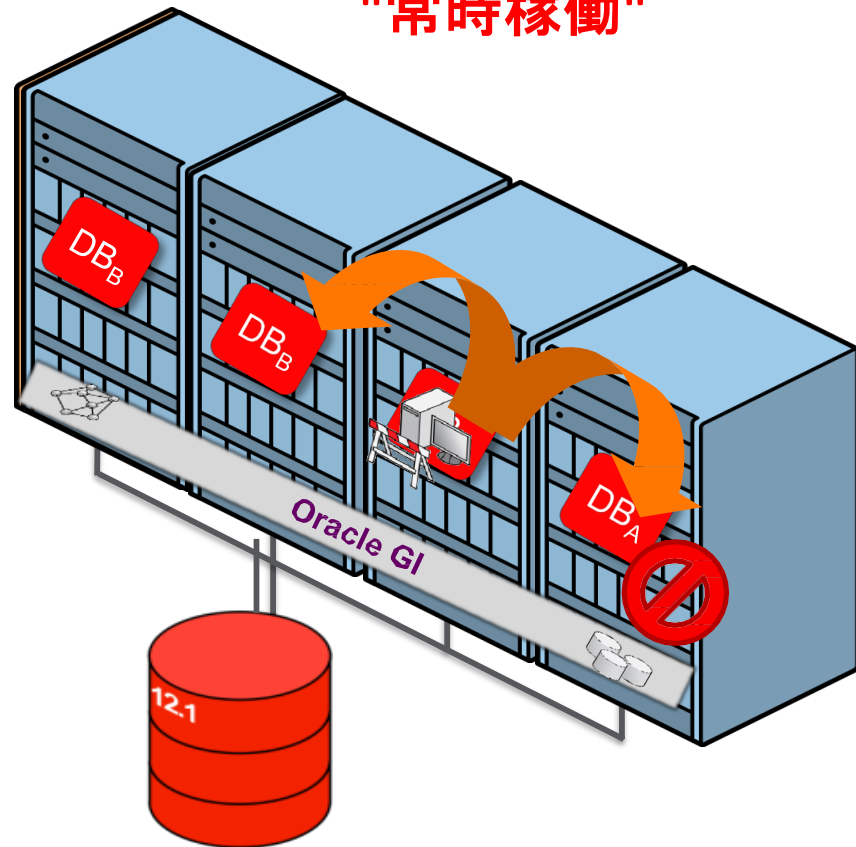
### シャーディング

- 大規模なOLTP
- スケジュールされたスイッチオーバー
- アクティブ/アクティブ・レプリケーション
- 異種

# Oracle Real Application Clusters

## 継続的な可用性

"常時稼働"



RACまたはRAC One Node

## Real Applicationのサービス・レベル

NEW IN  
18<sup>c</sup>

- 4000個のPDB、8000個のサービスのスケーリング
- オンライン・ローリング・メンテナンス
- 段階的なドレインおよび分散
- ほんの数秒間でDBをリカバリ
- **Application Continuity**が処理中の作業をリカバリ
- 常に一貫性のあるデータ

# Oracle Active Data Guard

## 継続的な可用性

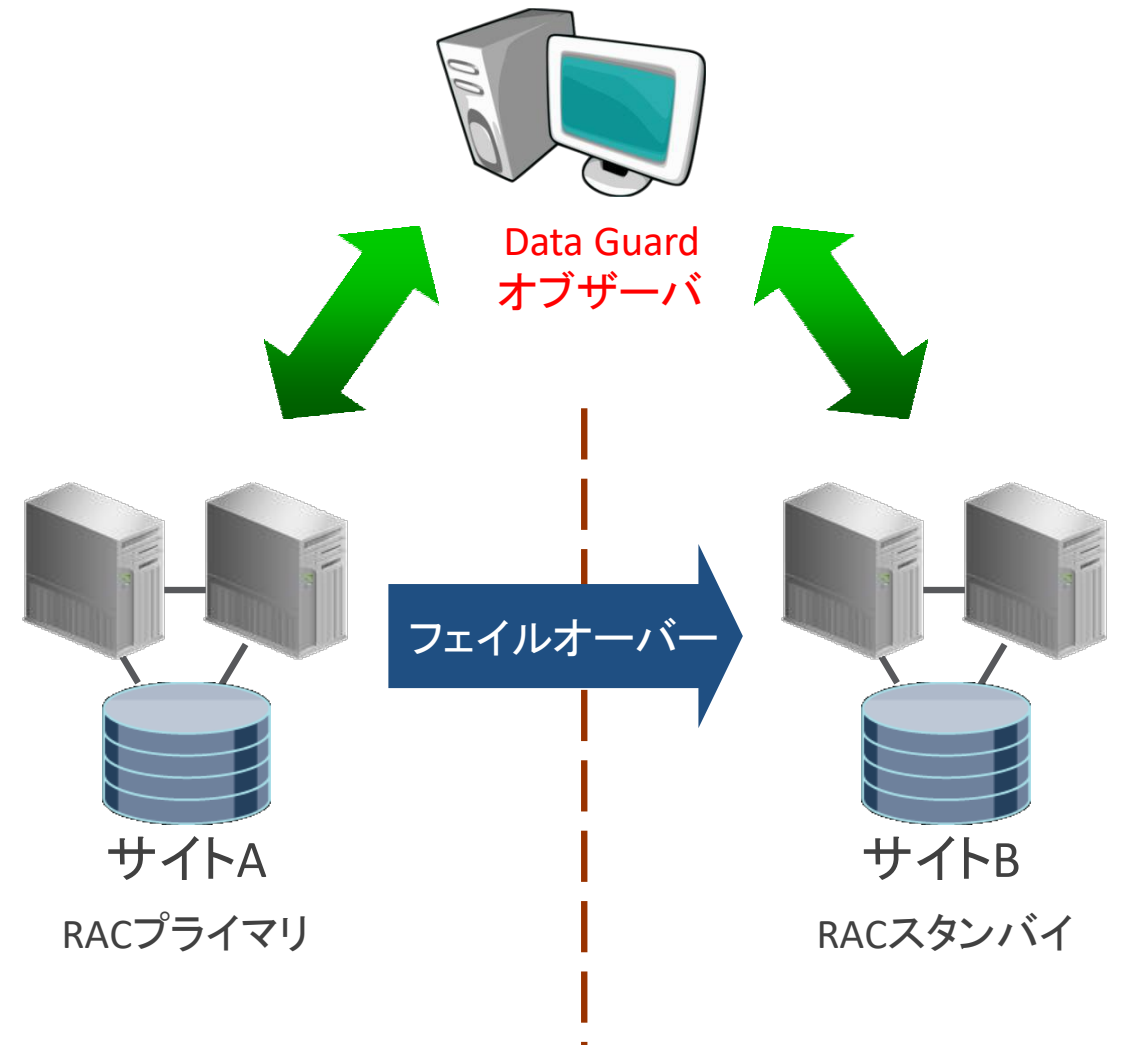
データ損失を防ぐためにローカルのDG-ADGにはFastSync/(no)affirm、リモート・サイトにはFarSyncを使用

アプリケーション・コンティニューイティによる処理中の作業のリカバリ

計画した場合はドレイン後にスイッチオーバー

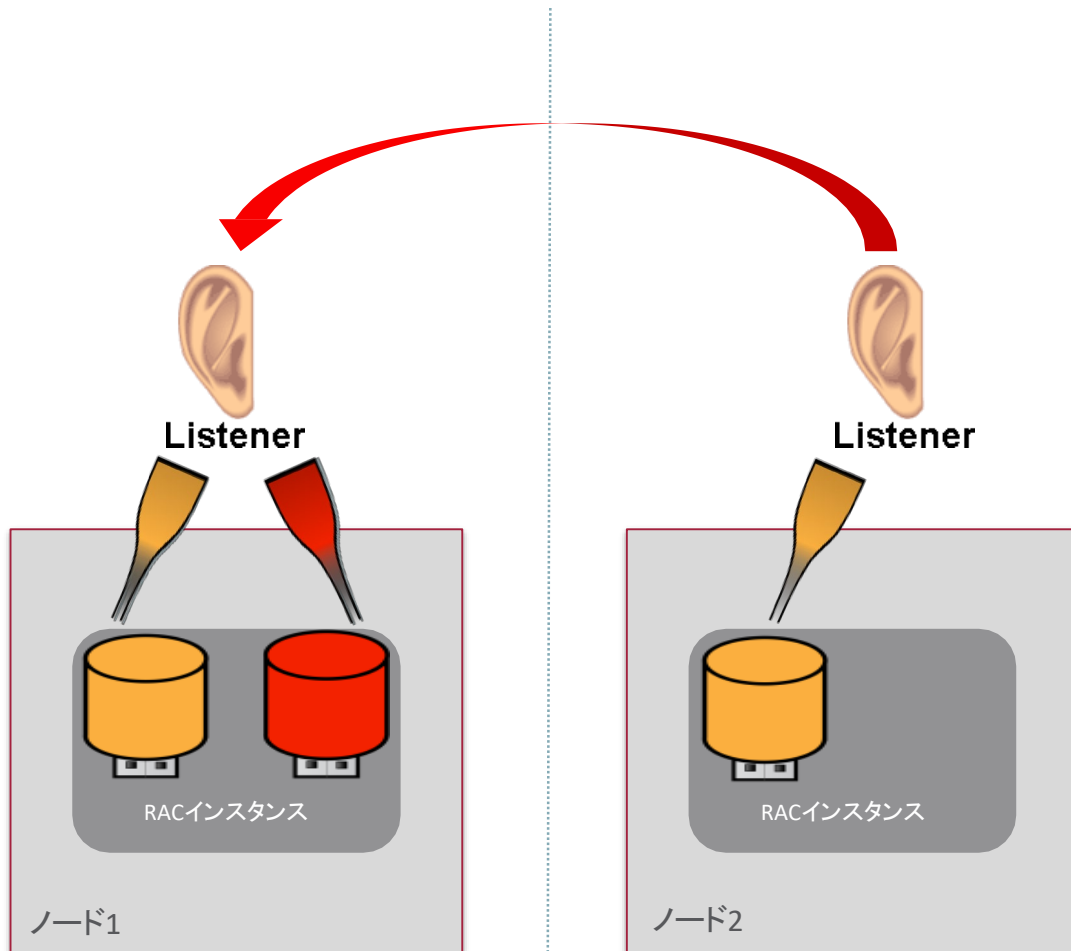
Switchover to <DG> [wait]

ADGセッションはロール変更後も継続



# 位置の透過性のためのサービス

サービスでアプリケーションの"ダイヤル・イン番号"が提供される



- 位置にかかわらず、アプリケーションは名前を維持
- 移動、再構築、優先順位付けによってサービスの提供方法を制御
- 分離されたバッチとOLTP
- 管理専用のDB名およびPDB名



# 接続は継続的であるように見える

12.2以降のすべてのドライバの標準

1つの場所で構成

単一の記述

```
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)))
```

自動再試行

```
(ADDRESS_LIST =  
  (LOAD_BALANCE=on)  
  ( ADDRESS = (PROTOCOL = TCP)(HOST=secondary-scan)(PORT=1521)))  
(CONNECT_DATA=(SERVICE_NAME = gold-cloud)))
```

DNSに依存しない

DB/PDB名ではないサービスを常に使用

# 瞬間的な中断に対するFAN

死んだものは死んだことを伝えることができない

## すべてのOracle製品でONSを使用

JDBC Universal Connection Pool

OCI/OCCIDライバ

ODP.NET Unmanaged Provider (OCI)

ODP.NET Managed Provider (C#)

OCIセッション・プール

WebLogic Active GridLink

Tuxedo

JDBCシン・ドライバ (12.2の新機能)

トラフィック・マネージャ・モードのCMAN (18cの新機能)



## 12cでの自動構成

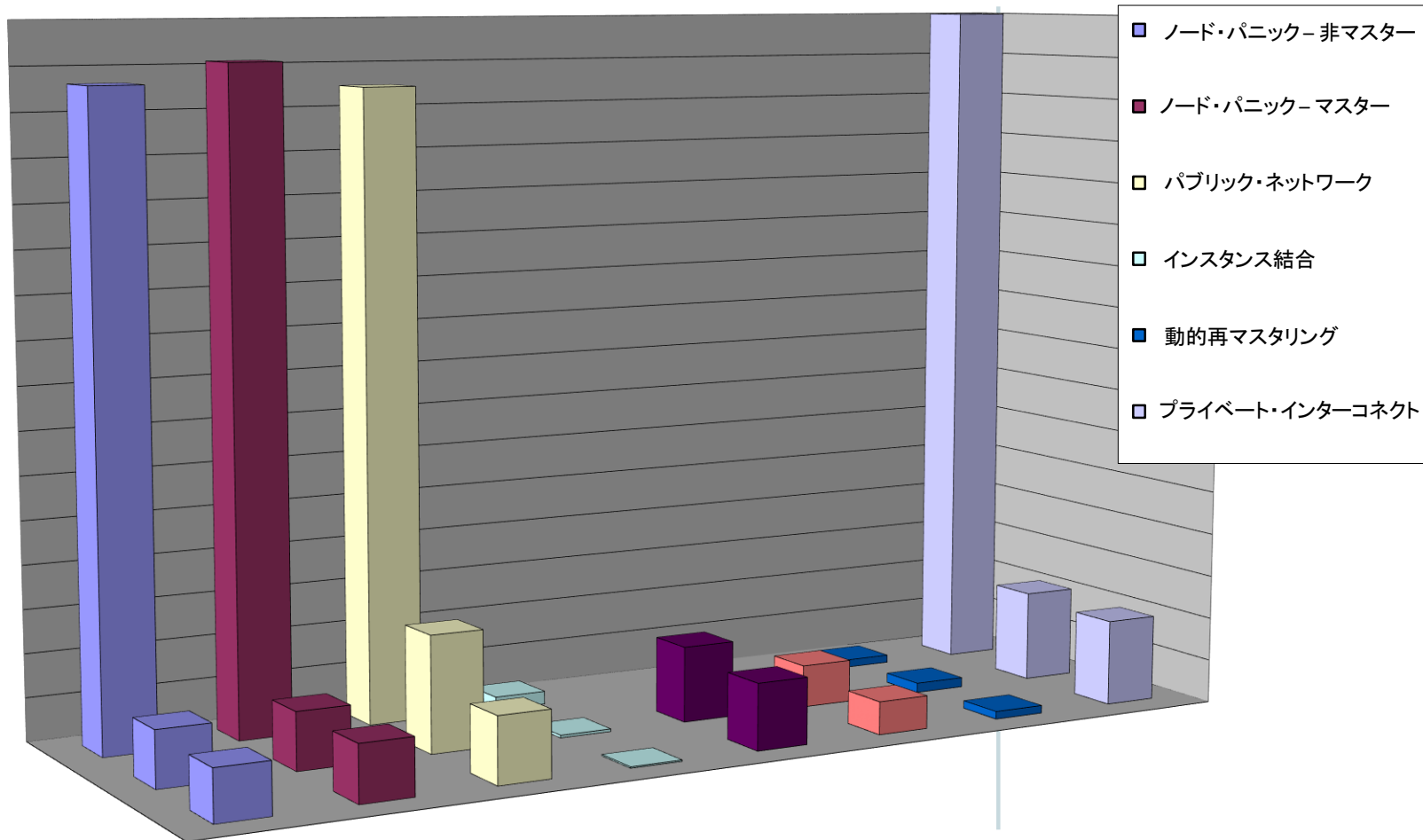
```
DESCRIPTION =
  (CONNECT_TIMEOUT=90)
  (RETRY_COUNT=20) (RETRY_DELAY=3)
  (TRANSPORT_CONNECT_TIMEOUT=3)
  (ADDRESS_LIST =
    (LOAD_BALANCE=ON)
    (ADDRESS = (PROTOCOL = TCP)
      (HOST=primary-scan) (PORT=1521)))
  ADDRESS_LIST =
    (LOAD_BALANCE=ON)
    (ADDRESS = (PROTOCOL = TCP)
      (HOST=second-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

ノード・セット1

ノード・セット2



# 適切なタイミングでのリカバリ



## チューニング可能なもの

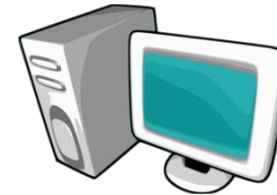
- ハードウェアおよび調整可能な misscount
- FAST\_START\_MTTR\_TARGET
- Oracle Flex ASM
- SCSIタイムアウト
- サービスの独立性18C
- PING\_TIMEOUT
- FSFOオブザーバ

# タイムアウトの調整

アプリケーション・タイムアウト(SLA)

アプリケーション・タイムアウト

- FSFO + クラッシュ・リカバリ
- 2 x MISSCOUNT + FSMT

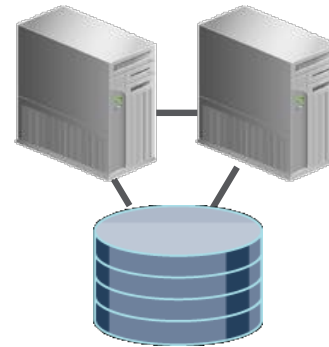


ファスト・スタート・フェイルオーバー (FSFO)

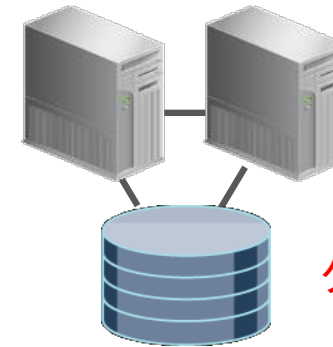
MISSCOUNT  
またはEXAスペシャル

ファスト・スタート  
MTTRターゲット  
(FSMT)

AC再実行タイムアウト



RACプライマリ

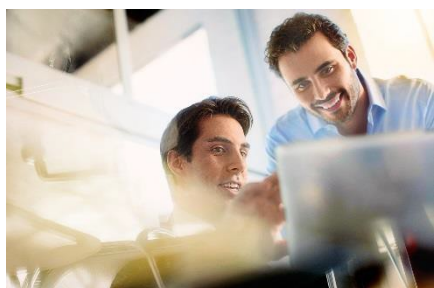


RACスタンバイ

クラッシュ・リカバリ  
(FSMT + オープン)

AC再実行タイムアウト

# お客様事例





# シカゴ・マーカンタイル取引所



# CMEグループの概要

CMEグループは、リスクを管理する必要がある人や組織、またはそれを受け入れることで利益を上げたい人や組織が集まる、世界トップレベルでもっとも多様性あるデリバティブ市場です。



- 複数の取引所を運営 – CME、CBOT、Nymex、COMEX
- 単一のプラットフォームで世界中の数百万の製品を取引
- 1日あたり平均1,560万件の契約量

- CMEによる決済 – すべての取引を照合して決済し、すべてのトランザクションの信用価値を保証
- 1,000兆ドルを超える価値の39億件を超える契約を決済
- もっともボリュームが大きかった日 – 選挙後に4,450万件の契約

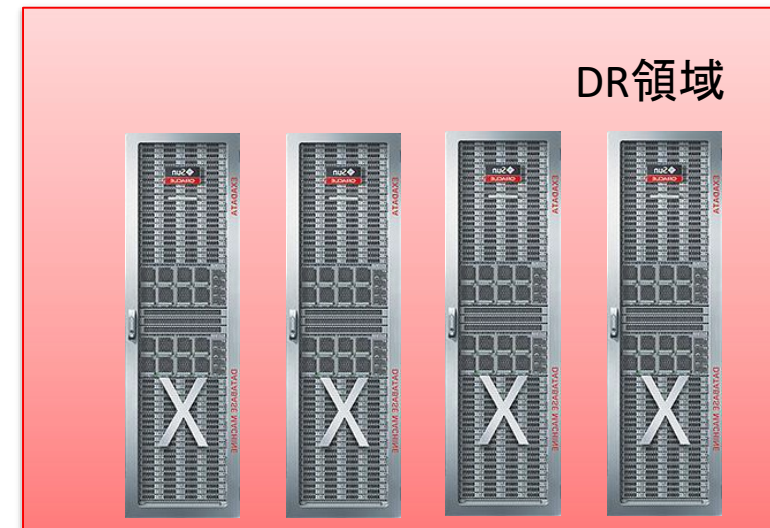
# CMEの高可用性の概要

## 要件

- 重要なDBの要件 – SLAに対して10秒
- コンポーネント障害によってDRイベントが発生しない
- 24時間365日アプリケーションが稼働
  - 計画メンテナンスを含む
- RPO – 30秒(災害のみ)
- RTO – 2時間(災害のみ)

## ソリューション

- Exadata
  - パフォーマンスに対応
  - 統合を許可
  - リカバリ時間を短縮(コンポーネント障害)
- Oracle Active Data Guard
- Application Continuity – 計画/計画外



# CMEがアプリケーション・コンティニュイティを採用する理由

- データベースの停止によって処理中の作業が失われる可能性がある
- スキーマ統合が原因となり、データベースの停止によって同時に多くのアプリケーションが影響を受ける可能性がある
- 重要なアプリケーションは24時間365日稼働するようになっている – これらは参照アプリケーション
- メンテナンス期間の短縮により、パッチ適用に代わるデータベースの計画停止時間をスケジュールするのが非常に難しい
- メンテナンス期間をデータベース・グループ専用にするのを避ける
- アプリケーションはシステムとして連携している。開始と正規化に数時間かかる場合がある



# 障害を防止するアプリケーションは困難

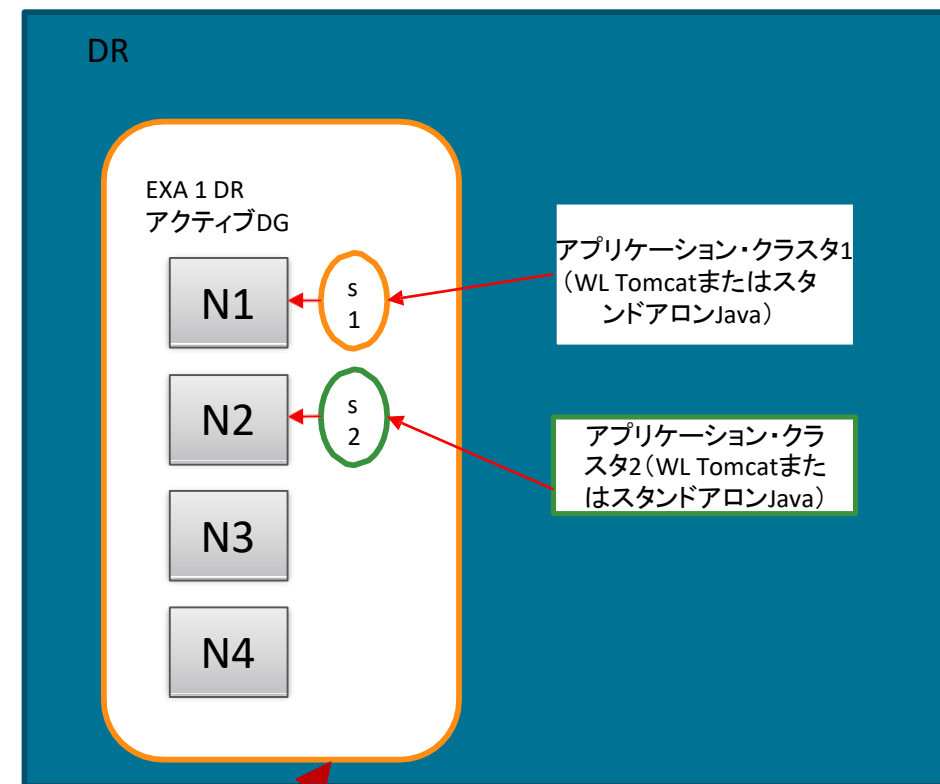
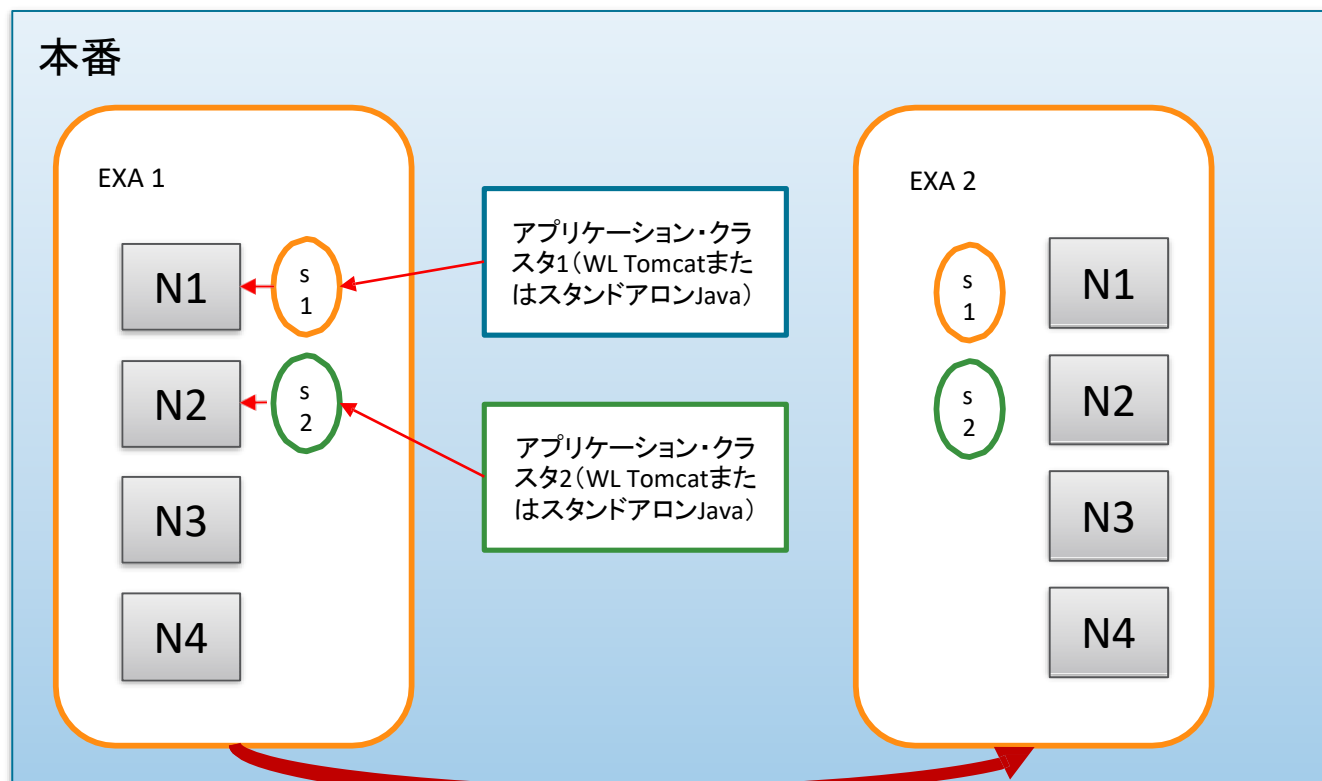
困難な点	今日の解決策
TCP/IPタイムアウトでの停止 – 前回の操作からの確認応答がないため、アプリケーションは問題を認識できない	<ul style="list-style-type: none"><li>• FAN – 高速アプリケーション通知</li><li>• FCF – 高速接続フェイルオーバー</li><li>• ONS – 通知サービス</li></ul> これらの機能が連携してTCPの停止を克服
障害発生後に存続しているノードまたはスタンバイ・データベースに再接続	アプリケーション・コンティニューイティで、接続文字列で構成可能なものすべての接続の再試行を自動的に実行
処理中のトランザクションがデータベースにコミットされることを保証	アプリケーション・コンティニューイティの機能でこれを透過的に処理。 トランザクションはクロスチェックされて安全に再実行される
信頼性により、データベースの計画メンテナンス中にアプリケーションの稼働状態を維持	ACはCMEにおいてレジリエンスがあることが分かっている



# 通常の操作

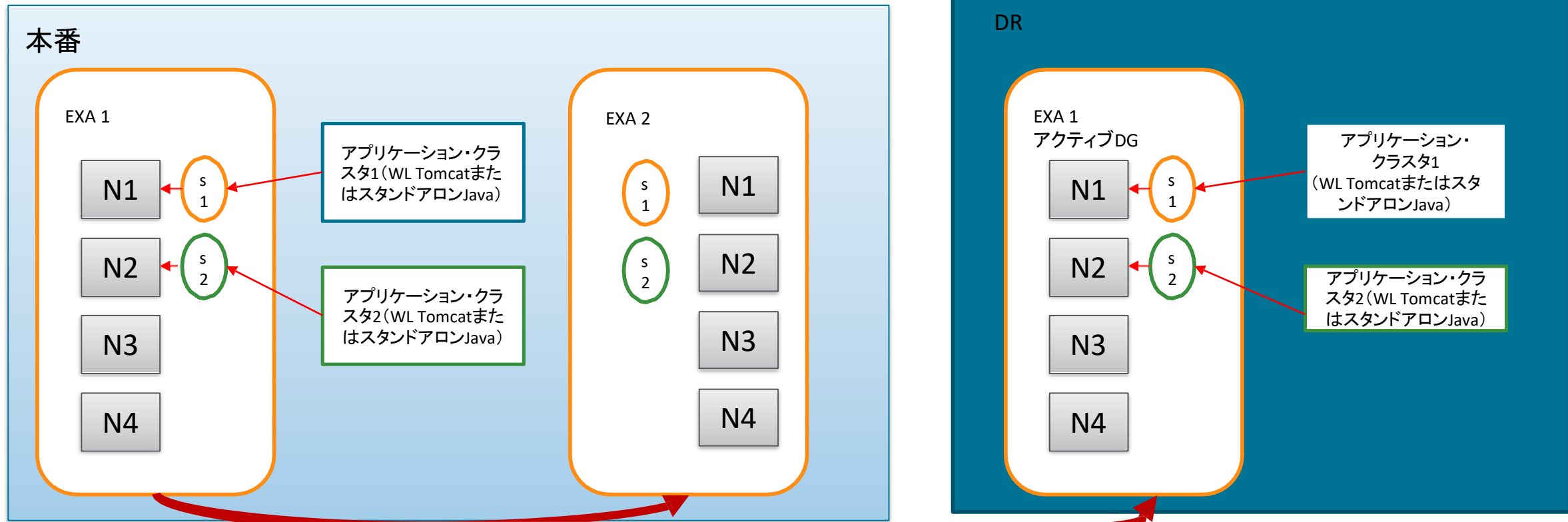
- すべてのOLTPサービスは、1つがアクティブで残りが使用可能として構成
- 環境全体で400を超えるサービス
- 100を超えるアプリケーション
- ノードの容量はアクティブに管理

- 多くのアプリケーション・サーバーは"待機状態"
- 重要なアプリケーションはROモードで接続



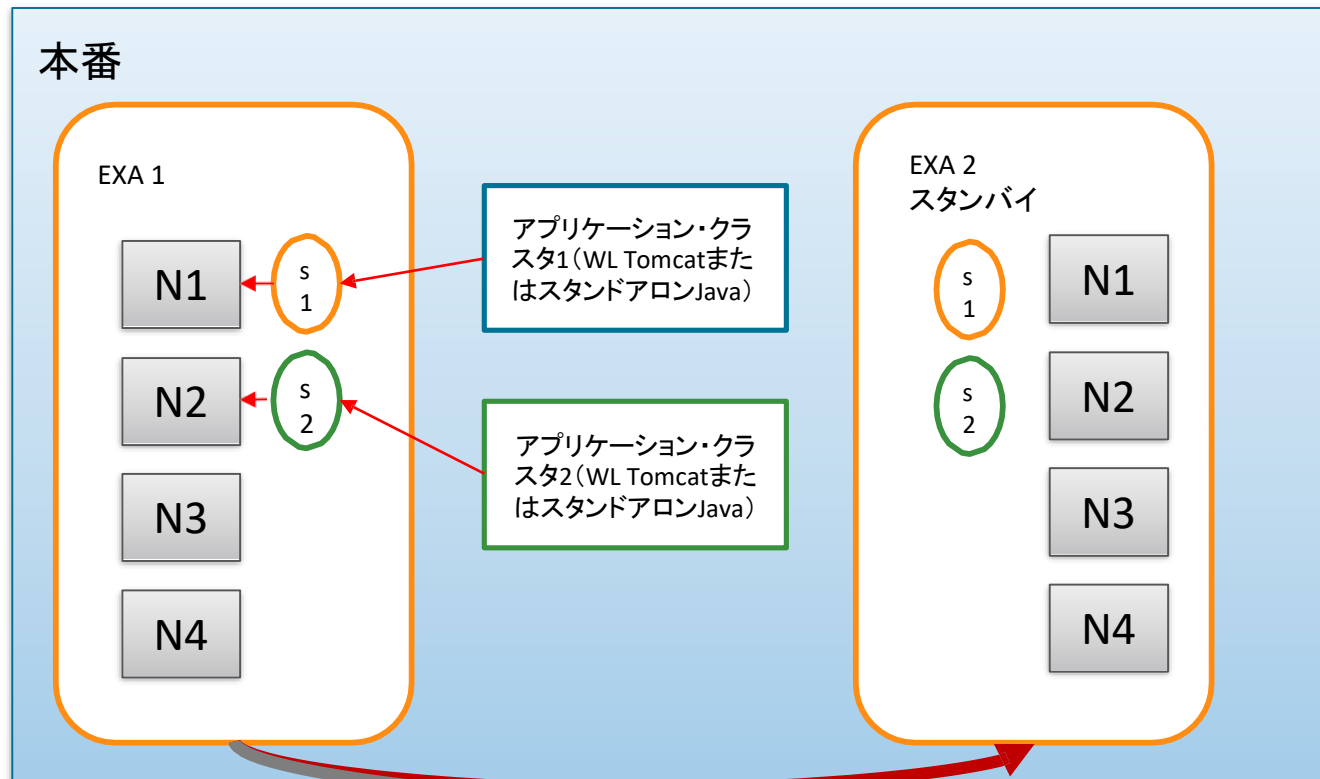
# 計画メンテナンス

- Exadataのフル・スタック・パッチ適用には4時間以上かかる
- CMEはローリング・パッチを行わない(期間が長すぎる)
- ACにより、パッチ適用中に、アプリケーションの稼働状態を保ち、更新を実施可能



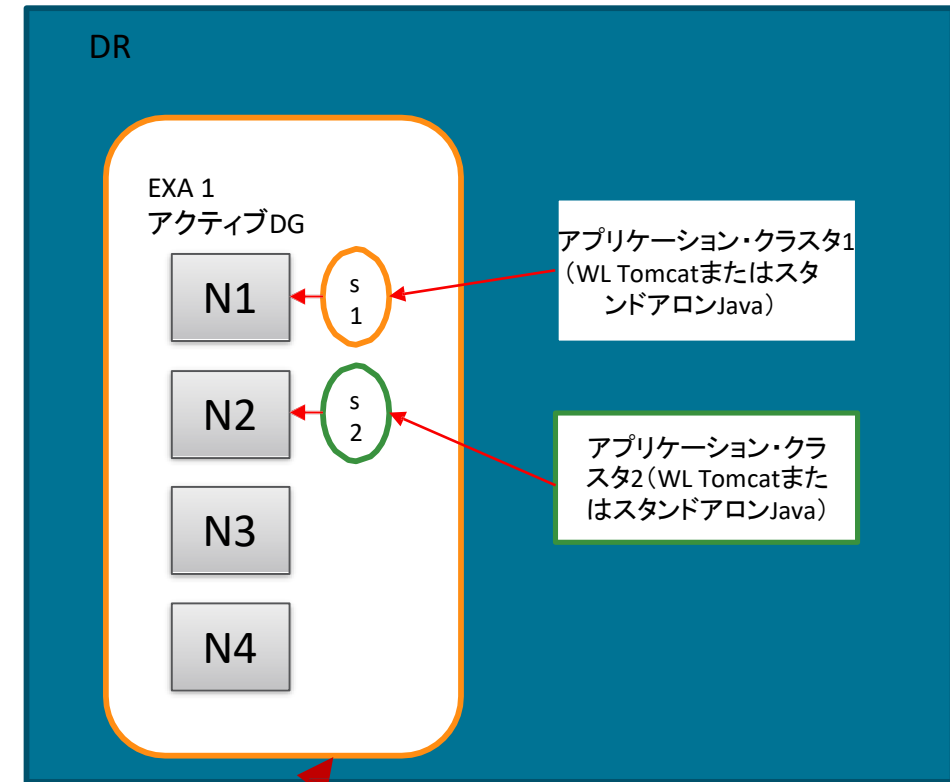
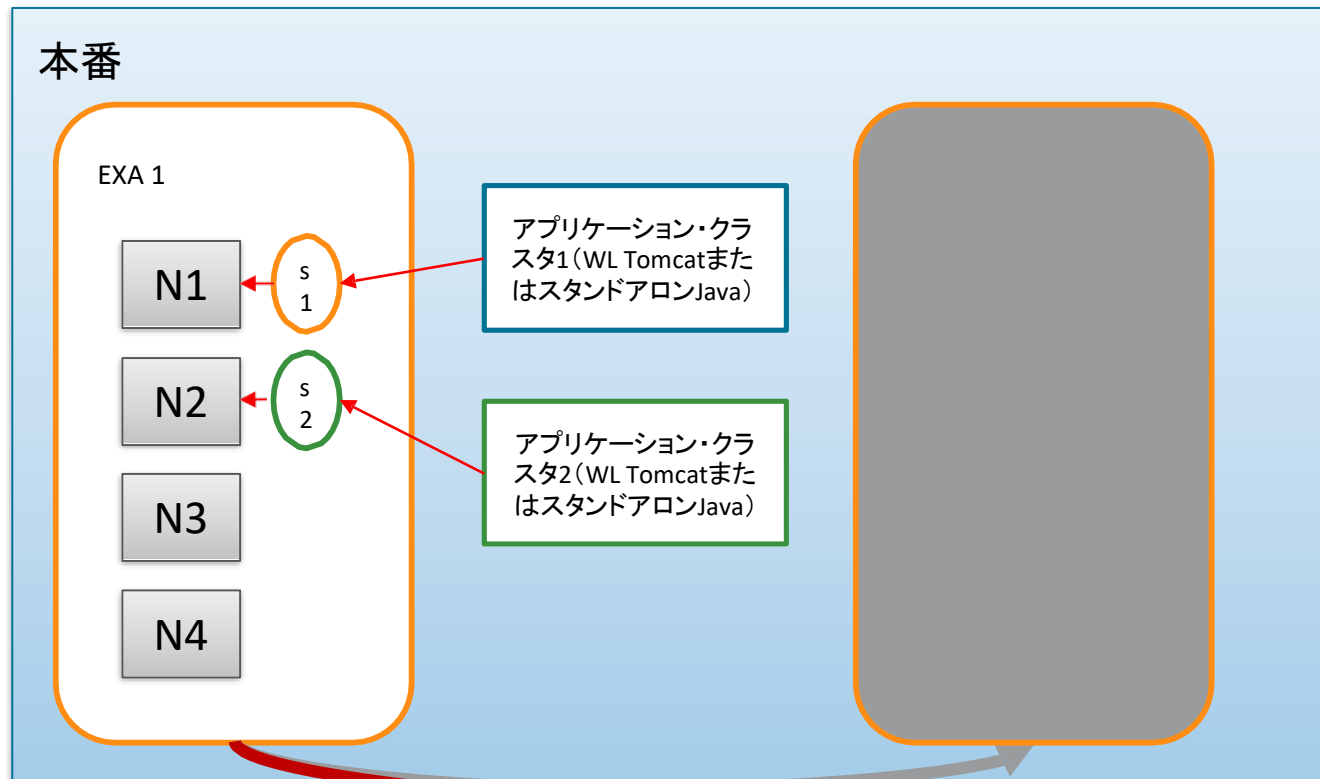
# 計画メンテナンス

- DRには常に最初にパッチが適用される
- DRのアプリケーションはオフラインになる
- 通常の変更期間が適用される
- 本番のアプリケーション変更はDRのパッチ適用と同時に実行される



# 計画メンテナンス

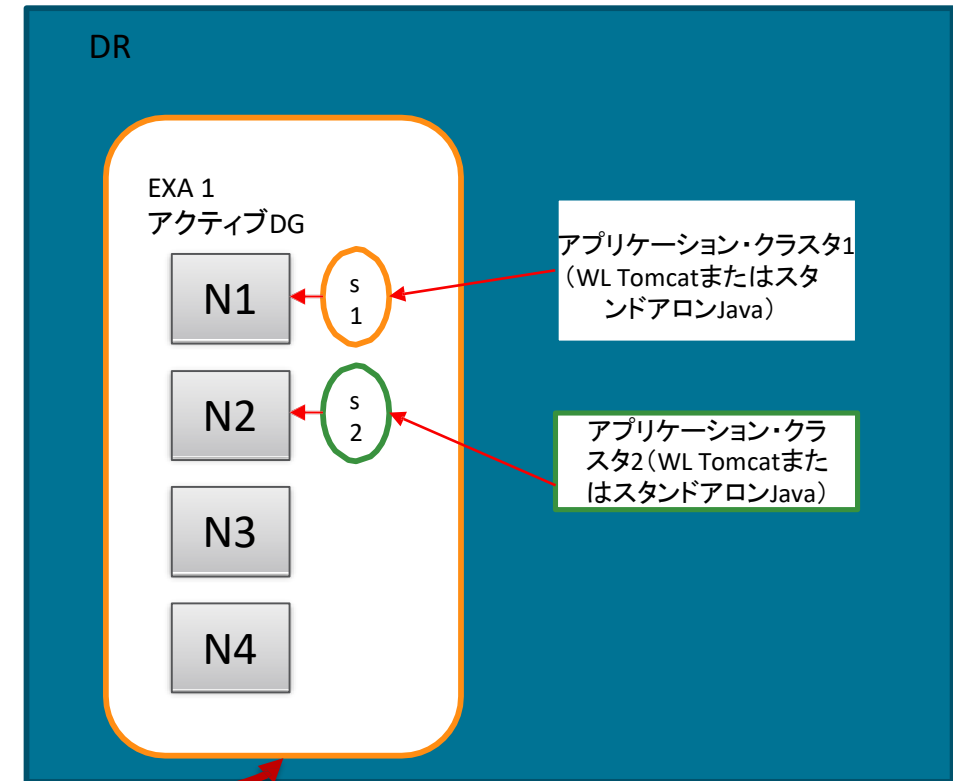
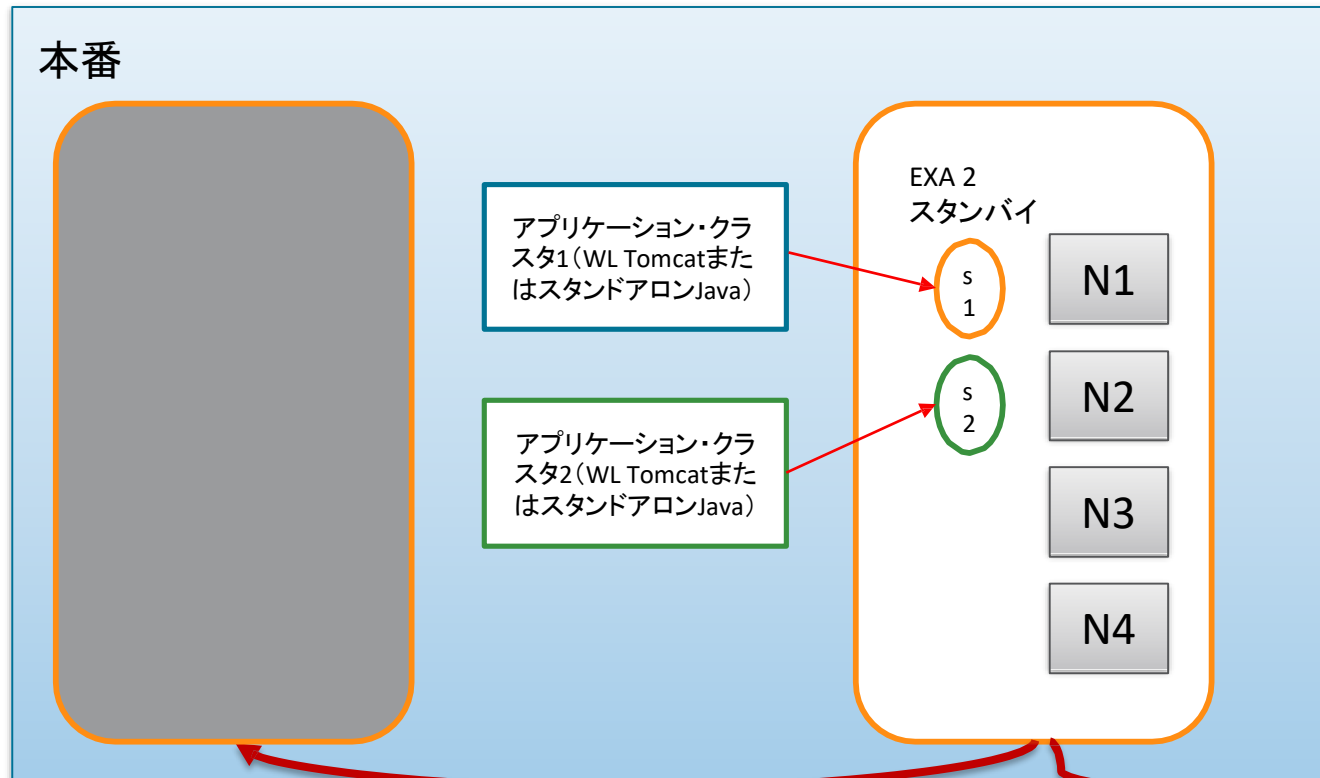
- ローカルのスタンバイ・データベースはDR後にパッチ適用される
- ローカルのスタンバイ・データベースへのパッチ適用は実行中のアプリケーションに影響しない
- 通常のメンテナンス期間中にパッチ適用される
- アプリケーションの変更とテストは継続可能



# 計画メンテナンス – Database As A Service

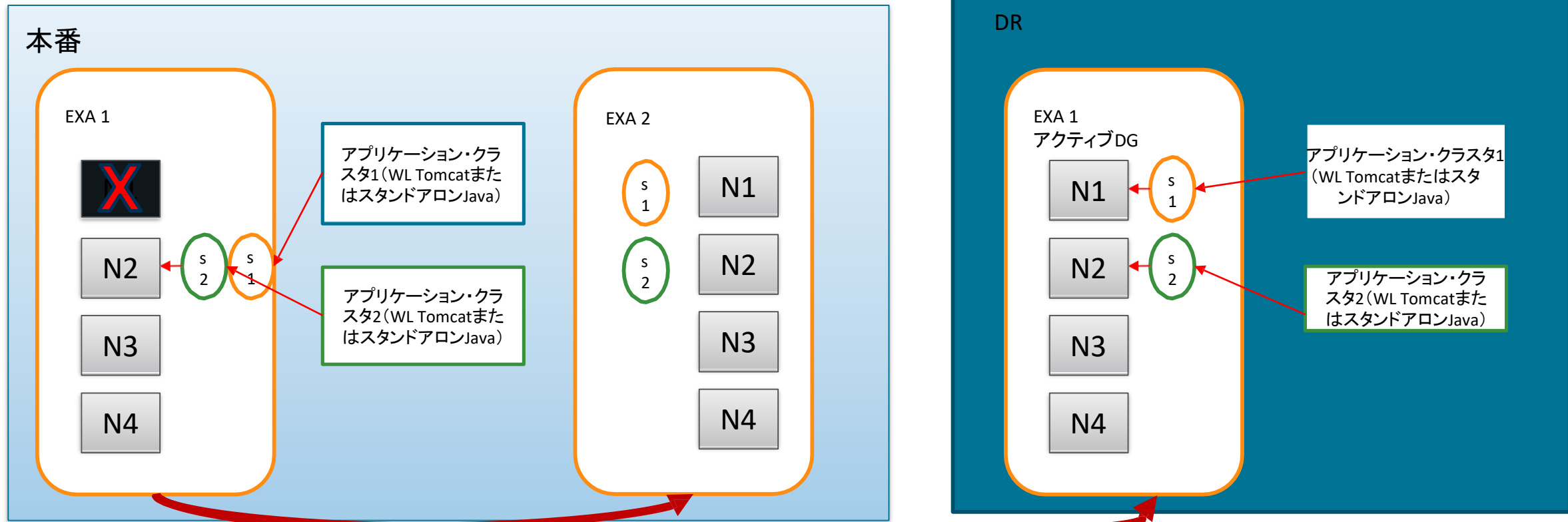
- AC準拠のアプリケーションは稼働して使用可能なままになる
- 非準拠のアプリケーションは停止して再開される (トランジション期間)
- データベース・スイッチオーバーが実行される
- LDAPジョブによって非準拠のアプリケーションの接続文字列が変更される

- 非準拠のアプリケーションが再開される
- メンテナンス期間中に変更とテストが継続される
- フェイルバックに対して処理が繰り返される



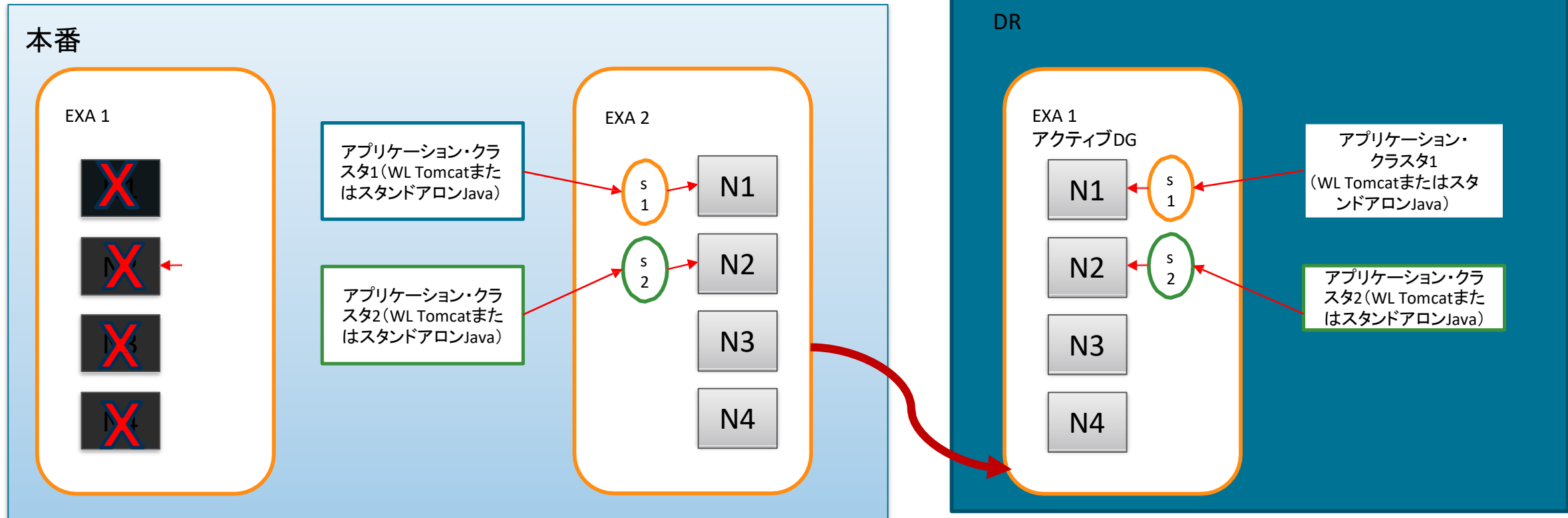
# 計画外停止

- ノード1で障害が発生
- 使用可能なインスタンスに対してすべてのサービスが失敗する(2に示します)
- アプリケーション・コンティニュイティを使用してアプリケーション接続がサービスの位置に従う



# 計画外停止

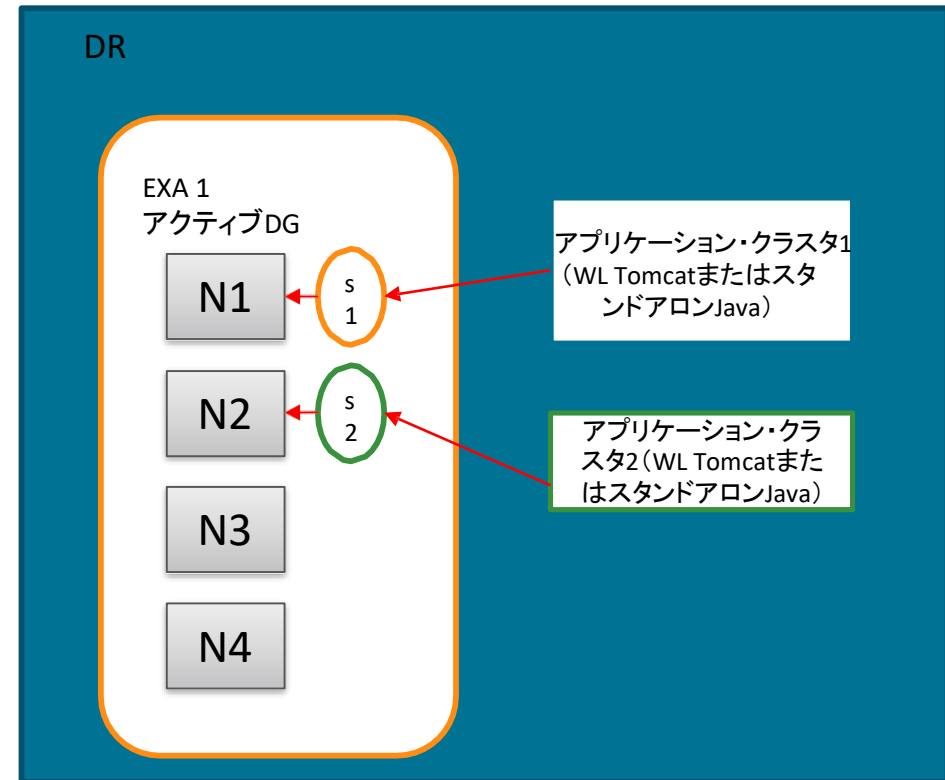
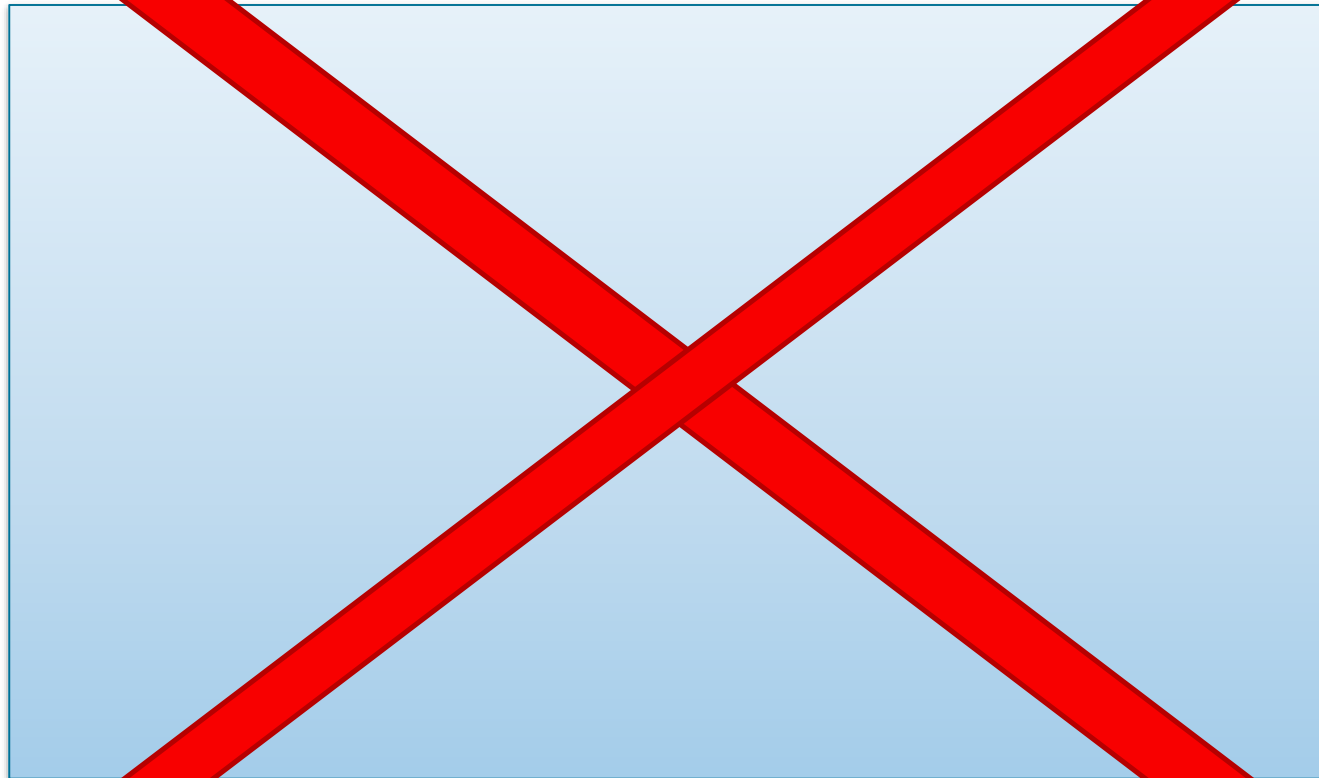
- Exadata全体で障害が発生した場合の状況
- CMEの場合 – これによってDRイベントが発生することは許可されない



## 計画外停止

- データセンターの致命的な障害
  - 制御不能なネットワーク停止(すべてのHAで障害が発生)
  - 建物の物理的損傷
  - EXA 1とEXA 2で同じ週に障害が発生

- お客様のROアクセスのために重要なアプリケーションが稼働
- データベースが変換される – アプリケーションでRWに変換
- すべてのアプリケーションが開始される – 2時間未満
- すべて自動化





## CMEのベスト・プラクティス

- 本番のミラーである最適なテスト環境
- 各々の変更速度に合った必要な間隔で行う、信頼性の高いHAおよびDRのテスト方法
- 現実的なテストのためのアプリケーションのシミュレーション
- 容量計画 – サーバーの使用率を50%未満に維持
- FANおよびFCFを使用して中断されるクライアント
- Application Continuityでサポートされる時間ベースのフェイルオーバー
- シングル・ポイント障害なし
- 強力な変更制御

