

# DB Replay (DataBase Replay) を使ったパフォーマンス・テストの概要手順

## DB Replay (Database Replay) の特徴と目的

アプリケーションからのデータベース操作の SQL コマンド実行記録を保存して、テスト環境にて状況復元操作として実行する

- ・ 負荷テスト
- ・ エラー再現テスト

## 本番データ処理の負荷の再現テストとは (DB Replay)

本番環境で**実行した全部の SQL 文**をテスト環境で実行し、**システム変更**の影響を検証します

このために、本番環境で実行した**すべての SQL 処理**を**ワークロード**にキャプチャして、テスト環境に移行して再現実行を行います

## 注意事項

テストを実行するためには、同じデータを用意する

すなわち、テスト前にバックアップを取得し、環境変更後のテストの実施前にはバックアップ・データをリストアして、SQL 文の結果が同じ処理となるようにして実行負荷の比較を行うこと

## 手順の概要

### 1. ワークロード・キャプチャー

DB Replay を使用して、本番システムで実行されているすべての SQL 処理をワークロードに保存 (ワークロードのキャプチャ)

### 2. テスト用のデータベースの準備

取得したワークロードの SQL が実行できるようなデータベース、およびデータの用意

### 3. リプレイ前処理

キャプチャした Database ワークロードからリプレイ (再現実行) 出来るファイルの作成 (リプレイ前の事前処理)

### 4. リプレイ&比較・分析

Database ワークロードのリプレイ実施

「キャプチャ時の処理」と「リプレイ時の処理」の比較分析レポートの作成

データベース・リプレイのタスク・リスト

番号	使用	タスク名	説明
1		本番ワークロードの取得	
1 - 1		ワークロードの取得	本番環境からワークロードを取得します 必要に応じてデータベース再起動を設定するためスケジュールすることができます
1 - 2		AWR データのエクスポート	AWR データをエクスポートして、取得したワークロードとリプレイしたワークロードの比較のパフォーマンスを改善します
1 - 3		ワークロード・ステージング領域へのコピー	後で前処理するため、取得したファイルの本番ステージング領域からワークロード・ステージング領域にコピーします クラスタ・データベースについては、複数のデータベース・インスタンスから取得したファイルをワークロード・ステージング領域で統合できます  <div style="border: 1px solid red; padding: 5px; margin-top: 10px;">                     RAC 環境で複数のデータベース・サーバーでクラスタ構成となっている環境では、SQL 実行をした DB サーバーごとにキャプチャファイルが取得される                      このため、キャプチャファイルが DB サーバーごとに分かれるのでこれを 1 つのフォルダに集める                 </div>
2		テスト・データベースの準備	
2 - 1		テスト用のインスタンスの生成	テスト用のデータベース・インスタンスの生成する
2 - 2		テスト用のデータの用意	ワークロード取得前のデータを、リストアを使って準備する
3		リプレイの準備	
3 - 1		ワークロードの事前処理	前処理は、取得したワークロードをリプレイする準備を行います この操作は取得したワークロードごとに 1 回実行する必要があります 前処理は、テスト・データベースで最適に実行されます

			<p>取得されたワークロードは、テスト・データベースからアクセス可能である必要があります</p> <p>〔 複数トランザクション処理の SQL 処理を取得するために、キャプチャ採取の時間帯を複数に分けて取得した場合には、事前処理済ワークロード（リプレイファイル）が別々のフォルダに分かれて作成されていることになるのでこれを1つのフォルダに集める 〕</p>
3 - 2		ワークロード・ステージング領域へのコピー	<p>前処理されたワークロードをワークロード・ステージング領域にコピーします</p> <p>前処理されたワークロード・ファイルは、データベース・サーバー、およびリプレイ中はリプレイ・クライアントからアクセス可能である必要があります</p>
3 - 3		リプレイ・クライアントのデプロイ	<p>リプレイ・クライアントを1つ以上のホスト・マシンにデプロイします</p> <p>リプレイ・クライアントは、前処理されたワークロードのリプレイに使用されます</p>
3 - 4		リプレイ・クライアント・ホストへのワークロードのコピー	<p>前処理されたワークロードを、1つ以上のリプレイ・クライアントにコピーする必要があります</p>
4		テスト・データベース上でワークロードをリプレイします	
4 - 1		ワークロード・リプレイ	<p>本番データベースのテスト・コピー版で前処理済ワークロードをリプレイします</p>
4 - 2		ワークロード・ステージング領域へのコピー	<p>リプレイの結果をワークロード・ステージング領域へコピーして、将来のリプレイと比較分析します</p> <p>〔 結果比較のためのレポートを出力させるためには、SQL 実行に関して出力された AWR スナップショット等の統計実績情報は、1つのディレクトリ・オブジェクトに集める必要があるのでファイルを1つのフォルダに集める 〕</p>
4 - 3		結果分析	<p>ワークロード・パフォーマンスにおける変更の影響を分析します</p> <ul style="list-style-type: none"> <li>・リプレイの期間比較レポート</li> <li>・期間比較 ADDM レポート</li> <li>・AWR の期間比較レポート</li> <li>・AWR レポート</li> </ul>

## DB Replay (Database Replay) の使用上の注意事項

- 個別パッチの確認

バージョンによって、DB Replay を使うにあたり適用する必要がある個別パッチがある場合がありますので、Oracle が公開している My Oracle Support の Note 情報を確認の上、必要な個別パッチを適用すること

- DB Replay のリプレイ操作で使用するファイルの配置

RAC 環境で DB Replay のリプレイを実行する際はプリプロセス後のファイル群を、すべてのインスタンスから参照できる共有ストレージ (ファイルシステム) 上に配置すること

- DB Replay キャプチャファイルの容量

DB Replay でワークロードのキャプチャをする場合には、キャプチャファイルの容量に注意すること

キャプチャ期間中のイベント「bytes received via SQL\*Net from client」のバイト数の 2~3 倍程度が目安

- DB Replay キャプチャ時の使用システム負荷

キャプチャ情報のファイル出力パフォーマンスは、キャプチャによるオーバーヘッドと関係するため、本番環境の処理に影響を与えないようにするために、キャプチャ出力先ストレージには極力速いレスポンスのストレージを用意すること

- DB Replay キャプチャ時の対象処理の限定

キャプチャフィルタを使用し、次のいずれかの設定を行うことで、キャプチャ/リプレイする対象を限定することができる

- リプレイしたいセッションのみを取得する
- リプレイしたくないセッションを除外する

すべてのセッションをキャプチャする場合でも、下記のセッションは本来のワークロード外となるため、除外するように設定する

※ (Enterprise Manager からキャプチャ作業を行うと、下記の 2 種類はデフォルトで除外されます)。

- Enterprise Manager 管理サーバー (OMS) からの管理用セッション
- Enterprise Manager エージェントの情報取得用セッション

- 正確なキャプチャを取得するためには  
完全で正確なキャプチャを取得するためには、ワークロードの取得前にデータベースを再起動すること
- データベースのレコード・データの更新  
リプレイでは実際にデータの更新も行われる。  
データと SQL に整合性があつていないと検証とならないため、キャプチャ前データの静止断面を RMAN や Data Pump などバックアップを取得し、リプレイを繰り返す場合は、リプレイ前にキャプチャ時点のデータにリストアすること
- PL/SQL プログラムのリプレイ時の注意点  
PL/SQL プログラムであるストアドプロシージャなどは、一つ一つの SQL がキャプチャされるわけではなく、コールのみがキャプチャされます  
PL/SQL プログラム内部に現在日時を呼ぶ SQL が含まれている場合など、キャプチャ時とリプレイ時で結果が異なる、ということが起こりえるため、注意する
- DB Replay の対象とならない処理  
DB Replay では、仕様上キャプチャされない処理や、キャプチャされた場合でもサポートされない処理が存在する（対象処理はバージョンによって異なる）  
例えば、
  - SQL\*Loader などのユーティリティを使用する、外部ファイルからのデータのダイレクトパスロード
  - PL/SQL 以外のアドバンストキューイング (AQ)
  - XA トランザクション
  - フラッシュバック問合せ※ サポートされない処理によってリプレイ中にエラーが発生する場合は、ワークロードからこれらを除外するようにキャプチャ時にフィルタ設定すること  
詳細はマニュアル（Oracle Database Real Application Testing ユーザーズ・ガイド 11g リリース 2、Oracle Database Testing ガイド 12c リリース 1）を参照のこと
- SYSTEM 表領域の必要容量  
リプレイ事前処理（プリプロセス）実行の際には、場合によって相当量の内部表を SYSTEM 表領域に一時的に作成するため、対象 DB の SYSTEM 表領域には十分な空き領域を確保しておくか、自動拡張可能にしておくこと