Projekterfahrungen mit Real Application Testing

Ulrike Schwinn Oracle Deutschland B.V. & Co. KG

Schlüsselwörter: Database Replay, SQL Performance Analyzer, Preprocessing, Capture, Replay, WRC, Erfahrungen, SAP, RAC, Mapping

Einleitung

Testen ist unerlässlich und sollte zu jedem Software-Entwicklungszyklus bzw. Qualitätssicherungsverfahren gehören. Testen ist allerdings auch aufwändig, der Aufbau der Testumgebung und das zur Verfügung stellen von adäquaten Testprozeduren kann etliche Personen-Tage bzw. -Monate kosten. Erschwerend kommt hinzu, dass man mit den meisten Testprozeduren nur eine Simulation durchführen kann, die in der Regel **nich**t dem realen Applikationscharakter entspricht. Mit **Oracle Database 11***g* steht eine neue Technik im Rahmen der Real Application Testing **Option** zur Verfügung, die eine einfache Lösung für diese Aufgabenstellungen bietet. Die Testabläufe zu vereinfachen und reale Applikationslast zu garantieren, zählen dabei zu den Hauptaufgaben.
Bei Real Application Testing handelt es sich - ganz einfach formuliert - um ein Werkzeug für die Datenbank, das einen Workload aufzeichnen und in einer Testumgebung wieder abspielen kann. Der Ausdruck Werkzeug ist dabei etwas irre-führend, da keine zusätzliche Installation einer separaten Werkzeug-Software für Real Application Testing notwendig ist. Die Nutzung erfolgt wie üblich über die Standardwerkzeuge Enterprise Manager oder PL/SQL bzw. SQL Aufrufe.

Welche Idee steckt dahinter? Ohne zusätzlichen Aufwand – wie Einsatz von speziellen Script Sprachen usw.- ist es möglich, eine Last bzw. einen SQL Workload aufzuzeichnen und in einer existierenden Testumgebung abzuspielen. Somit bietet sich Real Application Testing besonders an bei Datenbank Upgrades, Patch Anwendungen, Konfigurationsänderungen wie zum Beispiel dem Umstieg auf RAC oder ASM, Änderungen an Datenbank Sizing Parametern, Änderungen an der Hardware Plattform, am Betriebssystem oder am Storage, um nur einige Beispiele zu nennen. Folgende Fragestellungen sind dabei typisch: Funktioniert meine Applikation weiterhin wie gewohnt nach dem Upgrade auf ein neues Datenbank Release, nach dem Wechsel auf eine neuen Hardware oder nach Einspielung von Patches? Bleibt die SQL Statement Performance erhalten? Wie wirkt sich die Last gemäß I/O, CPU, Memory Metrik auf das neue System insgesamt oder gar pro Statement aus?

Real Application Testing beinhaltet zwei unabhängige sich ergänzende Lösungen, die je nach Charakteristik der Applikation bzw. Art der Analyse zum Einsatz kommen können. Es handelt sich dabei um das Database Replay und um den SQL Performance Analyzer. Auch hier wie bei vielen Produkten oder Features ist der Name bezeichnend für den Charakter und die Eigenschaften der beiden verschiedenen Funktionalitäten.

Dokumente bzw. Skripte, die die Funktionsweise und Verwendung von Real Application Testing erklären, gibt es in englischer und deutscher Sprache (siehe auch letzten Abschnitt). Daher sollen in diesem Artikel Projekterfahrungen und daraus resultierende „Best Practices“ behandelt werden.

Database Replay und SQL Performance Analyzer – Der Ablauf in Kürze

Database Replay (kurz DB Replay) zeichnet die gesamte Datenbanklast (auch Workload) unter Berücksichtigung aller konkurrierenden Zugriffe auf und führt eine Analyse bezogen auf die Gesamtlast der Datenbank durch. Das bedeutet, dass alle Datenbank Calls (Ausnahmen sind im Application Testing Handbuch vermerkt) im Workload enthalten sind. Der Ablauf sieht drei Verarbeitungsschritte vor:

- das Aufzeichnen auf dem Produktionssystem (auch Capture)
- eine spezielle Verarbeitung auf dem Testsystem (auch Processing), um die Abspielbarkeit der Dateien zu gewährleisten
- das eigentliche Abspielen auf dem Testsystem (auch Replay)

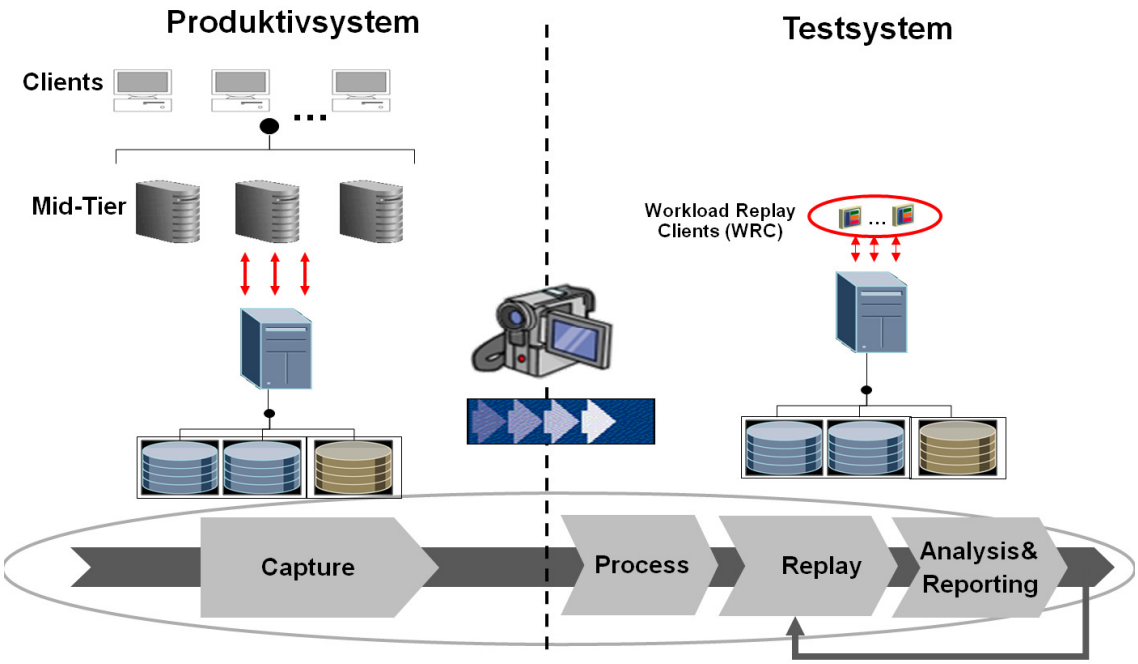
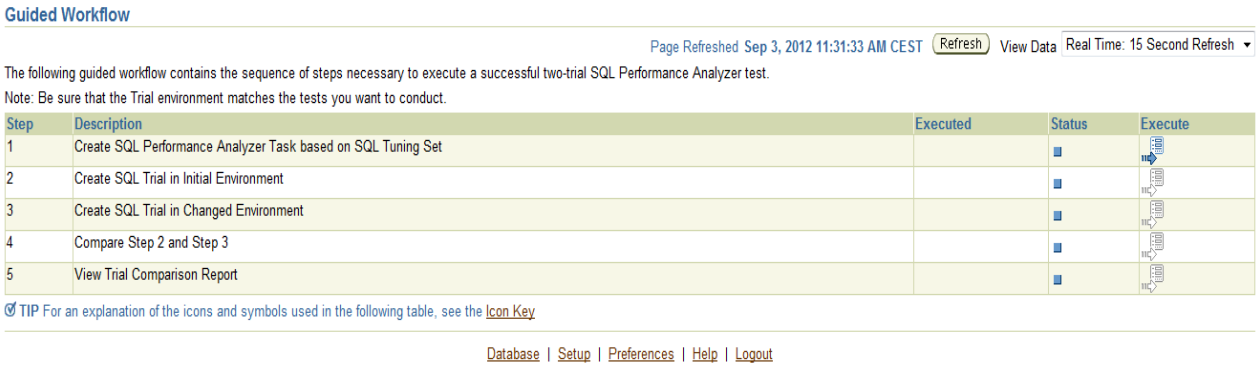


Abb. 1: DB Replay Ablauf

Der Fokus von **SQL Performance Analyzer** liegt auf der detaillierten Statement Analyse eines definierten SQL Workloads. Ein SQL Workload besteht dabei aus SELECT bzw. DML Statements (nur im Linemode), die über ein SQL Tuning Set (kurz STS) zur Verfügung gestellt werden. Der Ablauf sieht folgendermaßen aus:

- SQL Set generieren
- Erster Lauf
- Veränderungen durchführen
- Zweiter Lauf



.Abb 2: SPA Ablauf im Enterprise Manager

Das Ergebnis ist eine detaillierte Vergleichsanalyse - vor und nach der Veränderung - der einzelnen Statements gemessen an verschiedenen Metriken beispielsweise *elapsed time, cpu time* etc. Möchte man nun einen Tuning Prozess anschließen, ist dies einfach möglich durch die Integration des SQL Tuning Advisors oder durch die Nutzung von SQL Plan Baselines.

Tipps und Voraussetzungen
Auch hier gilt, wie bei allen Technologien, dass für die erfolgreiche Nutzung einige Regeln einzuhalten sind. Bevor Sie mit dem Testen starten, beachten Sie daher bitte folgende Hinweise:

1. Überprüfen Sie den Patch Stand Ihrer Produktions- und der Test-Datenbank. Der Patch Stand muss den Anforderungen aus der My Oracle Support Note (Doc ID 560977.1) entsprechen.

2. Überprüfen Sie, ob Ihr Capture Workload keine "non supported Features" beinhaltet (siehe Application Testing Handbuch).

3. Planen Sie einen ersten Testlauf für Capture, Preprocessing (oder Processing) und Replay ein. Sie erhalten erste Erfahrungen und bekommen einen Überblick über den Capture Umfang und über die möglichen Replay Optionen.

4. Begrenzen Sie die Capture Dauer auf wenige Stunden. Ideal sind 1 bis 2 Stunden.

5. Überprüfen Sie Ihr Testsystem auf Vollständigkeit (Datenbank User, Datenbank Tabellen usw.). Abgespielt wird nämlich immer, auch wenn Daten nicht korrekt bzw. unvollständig sind. Allerdings können dadurch sehr viele Fehler entstehen und damit das Replay Resultat wertlos machen.

6. Fahren Sie (wenn möglich) vor dem Capture die Datenbank herunter, um möglichst wenige sogenannte in-flight (noch offene) Transaktionen zu haben, die nicht aufgezeichnet werden können. Eine Kompromisslösung ist, das Capture bei geringer Last zu starten.

7. Nutzen Sie beim ersten Abspielen die gleichen Einstellungen (zum Beispiel Initialisierungsparameter) wie beim Capture.

8. Ziehen Sie das Tuning mit SPA in Betracht. Am besten tunen Sie vor dem Replay zuerst die Applikation mit SPA. Sie ersparen sich dadurch unter Umständen lange Abspielzeiten bzw. eine lange umständliche Ursachenforschung nach dem Replay.

9. Nutzen Sie Guaranteed (!) Restore Points für das DB Replay, um wieder einfach zu ihrem Testausgangspunkt zurückzukehren. Sorgen Sie dann allerdings dafür, dass ausreichend Platz für Flashback und Archive Log Dateien zur Verfügung steht.

10. Überlegen Sie, wie Sie mit externen Quellen wie DB Links, External Tables, URLs etc. umgehen - ob Sie diese zur Verfügung stellen können oder nicht.

11. Verwenden Sie vor dem Abspielen die Informationen aus dem Workload Analyzer Report - entweder über die graphische Oberfläche (ab 11.2.0.2) oder über den Linemode Aufruf (siehe auch Skript Download).

12. Planen Sie ausreichend Zeit für das Replay und die Nutzung verschiedener Replay Optionen ein.

Das Aufzeichnen (auch Capture)
Ein wichtiges Ergebnis aus allen durchgeführten Projekten vorab: Alle Aufzeichnungen konnten sehr einfach durchgeführt werden und haben in keinster Weise die Produktion beeinflusst!

Diese Erfahrungen stammen aus den unterschiedlichsten Projekten – von DWH Systemen mit Batch Jobs und Reports, bis zu OLTP Systemen mit hoher Transaktionslast und hoher Concurrency. Auch große Standard SAP Systeme waren unter den Testszenarios.

Auch wenn die Oracle Empfehlung lautet, die Datenbank vorab herunterzufahren, mussten die meisten Tests im **laufenden Betrieb** ohne jegliche Downtime der Produktion erfolgen. Das Einrichten von zusätzlichen Services im SAP RAC Umfeld (siehe Abschnitt unten) oder das Setzen des Parameters PRE_11g_ENABLE_CAPTURE bei 10*g* Datenbanken kann dabei auch im laufenden Betrieb erfolgen.

In unseren Tests wurden **Zeiträume** von 30 Minuten bis zu mehreren Stunden (z.B. auch 13 Stunden) aufgezeichnet. Die Applikationen zeichneten sich durch ganz unterschiedliche Lastprofile aus (8 Millionen User Calls in einer Stunde bis zu 600 Millionen User Calls in zwei Stunden). Die **Größenordnungen der Capture Dateien** lagen dabei unabhängig von der Länge der Aufzeichnung zwischen 50 und 300 GB. Die Größe der Capture Dateien steht dabei nicht im Zusammenhang mit dem Redo Aufkommen, sondern errechnet sich in Abhängigkeit der Statistiken rund um die Statements. Dabei spielen folgende Faktoren eine Rolle: Anzahl der verschiedenen Statements, Verwendung von LOBs, Bulk Binds, Anzahl der Connects, Einsatz von Parallelisierung etc. Die Größenordnung des Capture kann am einfachsten über ein kurzes Test Capture erfolgen. Möchte man den AWR Report zur Berechnung heranziehen, kann man sich an der Metrik „bytes received via SQL*Net from“ orientieren.

Aufzeichnen **im RAC Umfeld** funktioniert übrigens wie im Single Instance Umfeld. Kann man kein Shared Filesystem zur Verfügung stellen, können die Instanzen lokal aufzeichnen und müssen vor dem Preprocessing in ein gemeinsames Verzeichnis konsolidiert/kopiert werden.

Wichtige Voraussetzungen für das Capture sind ein leeres logisches Directory und **ausreichend Platz** auf einem Storage mit **guter Schreibperformance**. Zusätzlich sollte die Dauer bzw. die Menge des Capture begrenzt werden - ein Maximum von 4 Stunden bzw. eine Größenordnung, die unter 300 GB bleiben sollte, wäre sicher ratsam. Hintergrund ist einerseits die Handhabung beim Replay – ein kurzer Capture lässt sich immer leichter wiederholen und tunen als eine lange Aufzeichnung. Weitere Informationen finden Sie dazu auch in den Abschnitten zum Preprocessing und zum Replay.

Vielleicht noch ein Hinweis zum **Filtern**: Manchmal reicht es aus, sich auf bestimmte User bzw. Services beim Capture zu konzentrieren und mit einem Filterkriterium einzugrenzen. Umgekehrt kann es sinnvoll sein, die Aktivitäten von SYS und SYSMAN nicht zu capturen. Dies bringt den Vorteil, dass unter Umständen weniger aufgezeichnet wird und vielleicht sogar weniger Divergenzen und Fehler auf der Replay Seite entstehen.

Im **Single Instance SAP Umfeld** speziell auch beim Abspielen im RAC Umfeld, kann es zudem sinnvoll sein, vorab Services einzurichten. Damit kann man auf bestimmte Applikationsserver filtern bzw. nachher beim Abspielen im RAC Umfeld die entsprechenden Services auf die Knoten abbilden. Auch hier bedeutet dies datenbankseitig keine Downtime: Es müssen lediglich weitere Einträge an den SERVICE_NAMES Parameter in der init.ora/spfile angehängt werden bzw. zur Laufzeit per ALTER SYSTEM gesetzt werden. Auf Applikationsserver Seite muss die Konfiguration so angepasst werden, dass der entsprechende Applikationsserver nach dem Neustart den Connect gegen den neuen Service durchführt. Diese Konfigurationsänderung muss nicht zwingend zurückgenommen werden, sondern kann weiterhin beibehalten werden. Dabei ist es empfehlenswert, die Applikationsserver gleichmäßig auf die Services zu verteilen, so dass später auch eine sinnvolle Verteilung auf die RAC Knoten stattfinden kann. Eine sinnvolle Variante ist. einen Service pro SAP Applikationsserver zu definieren, damit man nachher völlig frei bei der späteren Verteilung auf die RAC Knoten ist.

Um am Ende eine guten **Überblick über das Capture** zu bekommen, kann man entweder einen Capture Bericht generieren oder eine Abfrage auf die View DBA_WORKLOAD_CAPTURES durchführen. So erhält man beispielsweise Informationen über die Capture ID, die Größe des Capture, Anzahl User Calls, aufgezeichnete DB Time und die Start SCN für den Aufbau des Testsystems.

```
select id, name, status, start_time, start_scn, dbtime*100/dbtime_total
dbzeit_in_prozent, user_calls, user_calls_unreplayable, errors,
awr_exported, duration_secs, filters_used, capture_size/1024/102
from dba_workload_captures where name like '&replayname';
```

Wenn in der Fehlerspalte ERRORS Fehler auftauchen ist dies übrigens kein Grund zur Beunruhigung und kann in der Regel ignoriert werden. Die Datenbankzeit in Prozent (hier DBZEIT_IN_PROZENT) gibt an,wie viel Datenbankzeit insgesamt aufgezeichnet worden ist. Da Background Operationen, beispielsweise die Nutzung von DBMS_SCHEDULER Jobs, nicht aufgezeichnet werden, kann diese Spalte in der Regel nie den Wert 100% erreichen.

Das Preprocessing

Das Preprocessing ist die Voraussetzung zum Abspielen des gesamten Workloads. Da diese Operation unabhängig vom Replay ist und nur einmal durchgeführt werden muss, kann dieser Vorgang auf einer **separaten Plattform** allerdings mit gleicher Datenbankversion (verglichen mit dem Replay System) erfolgen. Um die Performance zu erhöhen, gibt es die Möglichkeit einen Parallelitätsgrad beim Aufruf mitzugeben.

Ein **wichtiger Tipp** für das Preprocessing ist, ausreichend Platz im SYSTEM Tablespace zur Verfügung zu stellen, da dieser stark anwachsen kann. Sorgen Sie am besten dafür, dass kein Engpass entstehen kann.

Wie lange dauert nun ein Preprocessing? Da das Preprocessing die möglichen Replay Optionen vorbereitet, müssen die Daten in den Capture Dateien genau analysiert werden. Somit kann dieser Vorgang eine Stunde oder auch wie in einem unserer Tests 13 Stunden dauern. **Die Dauer des Preprocessing** hängt dabei von folgenden Faktoren ab:

- Anzahl und Größe der Capture Dateien
- Abhängigkeiten im Workload
- Genutzte Hardware und Storage

Das Preprocessing lässt sich dabei mit folgender Abfrage überwachen.

```
set serveroutput on
declare
  p number;
  z number;
begin
  p:=dbms_workload_replay.process_capture_completion;
  z:=dbms_workload_replay.process_capture_remaining_time;
  dbms_output.put_line('in Prozent:'||p);
  dbms_output.put_line('in Zeit:'||z);
end;
/
```

Ist das Preprocessing **erfolgreich** durchgeführt worden, ist die Spalte LAST_PROCESSED_VERSION aus DBA_WORKLOAD_CAPTURES mit der entsprechenden Versionsnummer gefüllt (z.B. 11.2.0.3.0). Das Ergebnis des Skripts liefert dann folgende Ausgabe.

```
in Prozent:100
in Zeit:
```

Es können allerdings auch `ORA-15590` Fehlermeldungen auftreten, wie im nächsten Beispiel zu sehen ist. Sie weisen nur darauf hin, dass einige Capture Dateien unvollständige Informationen enthalten. Beim Beenden des Capture sind Timeouts aufgetreten und somit konnten nicht alle aktiven Sessions ihre abschließenden Informationen in die Capture Dateien schreiben.

```
SQL> execute dbms_workload_replay.process_capture(capture_dir=>'PRE_DIR',
parallel_level=>6);
BEGIN dbms_workload_replay.process_capture(capture_dir=>'PRE_DIR',
parallel_level=>6); END;
*
ERROR at line 1:
ORA-15590: encountered incomplete workload capture files
ORA-06512: at "SYS.DBMS_WORKLOAD_REPLAY", line 1860
ORA-06512: at line 1
ORA-15590: encountered an incomplete workload capture file
```

Nach dem erfolgreichen Preprocessing ist das Verzeichnis der Capture Dateien um einige wenige Dateien bzw. Verzeichnisse angewachsen. Ein Beispiel: Erfolgt ein Preprocessing in der Version 11.2.0.3 wird ein neues Verzeichnis pp11.2.0.3.0 mit zusätzlichen Dateien erzeugt.

Wenn graphisch gearbeitet wird, sollte man unbedingt parallel dazu den **Workload Analyzer** anstoßen. Er gibt darüber Auskunft, welche Auffälligkeiten und Charakteristiken der Capture Load aufweist und mit welchen Optionen das Replay gestartet werden kann.. Die Linemode Variante des Workload Analyzers wird in den Skripten (siehe unten) zur Verfügung gestellt und sollte nach dem Preprocessing manuell gestartet werden.

Das Replay
Danach kann mit dem Replay begonnen werden. Hier müssen **verschiedene Entscheidungen** getroffen werden. Soll zum Beispiel nachträglich gefiltert werden, wie viele WRC Clients müssen gestartet werden, und welche Optionen sollen beim Replay verwendet werden. Die ersten beiden Fragen lassen sich recht leicht beantworten. Ob nachträglich gefiltert werden soll, hängt vom weiteren geplanten Testverlauf ab. Die **Anzahl der WRC Clients** lässt sich sehr leicht entweder über das WRC Client Werkzeug selbst oder über folgenden Aufruf des Package DBMS_WORKLOAD_REPLAY bestimmen:

```
set serveroutput on
declare
result varchar2(4000);
begin
      result :=  dbms_workload_replay.calibrate('&replaydir');
      dbms_output.put_line(result);
end;
/
```

Die Ausgabe ist in XML und sehr nützlich. So wird nicht nur die Anzahl der WRC Clients aufgelistet, sondern auch Informationen aus DBA_WORKLOAD_CAPTURES zum aktuellen Capture und Preprocessing Status zusammengefasst. Die WRC Clients können auf dem Testrechner selbst oder unabhängig davon auf beliebigen **zusätzlichen Testrechnern** gestartet werden. Dazu ist keine Oracle Software Client Installation notwendig. Es reicht aus, die WRC Instant Client Software von OTN (zum Beispiel „Instant Client Downloads for Linux x86-64“ unter http://www.oracle.com/technetwork/topics/linuxx86-64soft-092277.html) auf den entsprechenden oder die entsprechenden Server zu kopieren. Wichtig ist, dass alle WRC Clients Zugriff auf alle Capture Dateien haben.

Während des Aufzeichnens werden auch die Connection Strings aufgezeichnet. Möchte man auf dem Replay System ein erfolgreiches Abspielen gewährleisten, muss unter Umständen ein **Mapping der Connections** erfolgen. Für RAC Datenbanken kann man zum Beispiel alle Connection Strings auf einen Load Balancing Connection String abbilden. Ein anderer Anwendungsfall wäre das Umleiten eines speziellen Workloads auf eine spezielle Instance bzw. auf einen speziellen Service. Im SAP Umfeld wird das Mapping beim Wechsel von Single Instance auf RAC empfohlen – wie im Kapitel „Das Aufzeichnen“ schon ausführlich beschrieben wurde. Folgendes Beispiel zeigt eine Verwendung. Ein Mapping der Connection ID 101 könnte dann folgendermaßen aussehen:

```
execute DBMS_WORKLOAD_REPLAY.REMAP_CONNECTION (connection_id => 101,
replay_connection => '<host>:<port>/SERVICE1');
```

Dabei kann man für den *replay_connection* Parameter einen beliebigen Connect Identifier (wie Net

Service Name, Database Service oder Net Service Aliase) und eine beliebige Naming Methode wählen. Um das aktuelle Mapping zu überprüfen, verwenden Sie die View DBA_WORKLOAD_CONNECTION_MAP. Sie gibt Auskunft über die Connection und Replay IDs und den zugehörigen Connection Strings.

Bevor man das Replay startet, muss über die Art des Replays nachgedacht werden. Zur Erinnerung: Der Parameter *synchronization* kann den Wert TRUE (Defaulteinstellung), FALSE oder OBJECT_ID haben. TRUE oder FALSE schalten die COMMIT Synchronisierung (auch preserve commit order) ein bzw. aus. Ist die Synchronisierung mit TRUE eingeschaltet, wird die COMMIT Reihenfolge des Capture Workloads beibehalten. Jede Aktion wird erst dann abgespielt, falls alle COMMIT Aktionen, die vorher ausgeführt worden sind, beendet sind. Dies kann natürlich zu einer Verzögerung beim Abspielen führen (siehe auch Metrik *elapsed time*). Die Metriken *db time*, *cpu* usw. bleiben davon unberührt. Allerdings kann es zu Irritationen bei der Testauswertung kommen, wenn der Replay im Extremfall ein Vielfaches der Capture Dauer beträgt. Nutzt man hingegen den Wert FALSE kann es zu Divergenzen kommen. Sind diese gering, dann spricht natürlich nichts gegen diese Art des Replays, wie wir in mehreren Tests beobachten konnten. Der Wert OBJECT_ID hingegen bedeutet, jede Aktion wird dann abgespielt, wenn die relevanten COMMIT Aktionen beendet sind. Die relevanten COMMIT Aktionen sind diejenigen, die mindestens ein Datenbank Objekt verändert haben, welches durch die gegebene Aktion nun referenziert wird. Diese Art der Synchronisierung sollte unbedingt bei der Wahl der Replay Optionen berücksichtigt werden.

Nun können die entsprechenden Reports generiert werden, um eine Bewertung des Replay Laufs durchzuführen. Alle Reports lassen sich im Linemode generieren und stehen häufig in TEXT, HTML oder XML Format zur Verfügung. Einfacher zu erzeugen sind diese natürlich über die graphische Oberfläche. Generell sind folgende Reports möglich. Sie liefern dabei unterschiedliche Bewertungsmaßstäbe.

- **Replay Report** gibt erste Informationen zu Replay Statistiken, Top Events, Workload Profil und Divergenzen.
- **Divergence Report** listet die Divergenzen im Detail.
- **Compare Period Report** gibt erste Informationen zu der Performance im Vergleich zum Capture oder einem weiteren Replay.
- **AWR Difference Report** bzw. **AWR Report des Capture** und des **Replays**
- **ASH Report**

Die letzten beiden Berichtsarten sind gängige und bekannte Mittel beim Datenbank Tuning. Die ersten drei hingegen sind typische Real Application Testing Reports. Konzentrieren Sie sich zuerst auf den Replay Report (auch vergleichbar mit dem Capture Report). Dieser liefert die ersten Informationen zu den Replay Statistiken, den Divergenzen und den Top Events auf. Informationen zu Performance und genauere Angaben zu Divergenzen erhält man über die speziellen Reports - den Compare Period Report bzw. den Divergence Report. Ein Blick in den **Divergence Report** kann manchmal helfen, Grundursachen für eine hohe Divergence Statistik zu erhalten. Ein Beispiel wäre eine Liste von Statements, die wegen fehlender User oder Zugriffsrechte nicht ausgeführt werden können. Die Ursache kann dann schnell behoben werden und ein weiterer Replay Lauf kann erfolgen. Der **Compare Period Report** ist für eine erste Performance Einschätzung sehr wertvoll. Im Unterschied zu den AWR und ASH Reports werden weniger Details und Metriken der Performance Analyse aufgezeigt, dafür wird allerdings ein guter Überblick über wichtige entscheidende Performance Faktoren gegeben. Es werden Optimizer- und Memory-Einstellungen, wichtige Initialisierungparameter, wichtige Performance Statistiken und Top Statements miteinander verglichen

und zusätzlich einige Hardware Statistiken ausgegeben wie I/O bzw CPU Nutzung. Am Schluss werden noch die Ergebnisse des ADDM Laufs aufgelistet.

Bei der Bewertung der Reports sollten Sie allerdings berücksichtigen, was und wie sie aufgezeichnet bzw. abgespielt haben. So kann beispielsweise ein Capture Workload mit „warm gelaufener" Datenbank andere I/O Zeiten aufweisen als ein Replay Lauf.

Die Nutzung von SQL Performance Analyzer
Der Fokus von SPA liegt, wie eingangs schon erwähnt auf, der detaillierten Statement Analyse eines definierten SQL Workloads. Welche Statements können von SPA analysiert werden? Die Antwort ist sehr einfach: Alle Statements, die einen Ausführungsplan besitzen, können mit SPA verarbeitet werden. Es gibt auch hier einige Einschränkungen, die dann im SPA Lauf als ERROR oder UNSUPPORTED gelistet werden.

SPA ist unabhängig von DB Replay nutzbar ist und eine gute Ergänzung bzw. Vorbereitung für den DB Replay Testlauf. So kann zum Beispiel **während eines Replay Laufs** automatisch ein SQL Tuning Set aufgezeichnet werden, um danach gleich zur Statement Analyse überzugehen. SPA listet nicht nur die unterschiedlichen Metriken vor und nach der Veränderung auf, sondern gibt auch den Ausführungsplan an und den Anteil des Statements am gesamten Workload. Dabei werden die Statements nach verschiedenen **Kategorien** eingeteilt: *improved*, *regressed*, *unchanged*, *changed_plans*, *errors* usw. Auf diese Weise kann man sich beim Tuning sehr einfach den kritischen Statements mit verändertem Ausführungsplänen widmen. So konnte in einem unserer Projekte die Performance von kritischen Statements um 90% durch Einsatz von SQL Plan Baselines gesteigert werden. Zur Verifizierung wurde danach noch einmal ein Database Replay Lauf durchgeführt, um die Auswirkung auf die Gesamt-Performance zu messen.

Ein SPA Lauf kann aber auch **unabhängig vom Database Replay** Lauf verwendet werden. Der Vorteil ist die einfache Handhabung und die **Verfügbarkeit**. Da keine DML-Statements die Testumgebung verändern - DML Statements werden automatisch zurück gerollt - ist kein Zurücksetzen nach dem Test erforderlich. Kennt man die Top Statements, deren Performance beibehalten bzw. verbessert werden soll, kann man auch nur diese testen, ohne den gesamten Workload abspielen zu müssen.

Fazit
Unsere Projekte konnten zeigen, wie wertvoll der Einsatz von Real Application Testing sein kann. So gibt es zum Beispiel Kunden, die eine Änderung oder Patch Einspielung erst dann freigeben, wenn ein erfolgreichen Database Replay Lauf durchgeführt werden konnte.

Folgende Ergebnisse unserer Projekte geben noch einmal wichtige Aspekte bei dem Einsatz von Database Replay wider:

- Das Capture stellt keine Beeinflussung der Produktion dar.
- Die Tests verhelfen zu **Sicherheit und Vertrauen** für Migrationen/Upgrade.
- Es wurde ein besseres **Verständnis für eigene Applikationen** gewonnen.
- Die Tests können **entscheidende Argumente** bei Plattformwechsel liefern.

Um die Technologie richtig einzusetzen, ist allerdings, wie bei allen neuen Techniken eine gewisse Einarbeitung notwendig. Stehen die entsprechenden Skripte erst einmal zur Verfügung und ist ein gewisses Verständnis für die Abläufe vorhanden, ist die Nutzung recht einfach. Wichtig ist dabei bestimmte Richtlinien, die im Kapitel „Tipps und Voraussetzungen" gelistet sind, einzuhalten.

Bleibt abzuwarten welche weiteren Features in den nächsten Releases implementiert werden. Unabhängig davon kann sich schon jetzt der Einsatz lohnen.

Weitere Informationen

Beispielskripte für Database Replay
https://apex.oracle.com/pls/apex/GERMAN_COMMUNITIES.SHOW_RESOURCE_BY_FNAME?P_TIPP_ID=221&P_FILE_NAME=rat_akt.zip

Beispielskripte für SQL Performance Analyzer
https://apex.oracle.com/pls/apex/GERMAN_COMMUNITIES.SHOW_RESOURCE_BY_FNAME?P_TIPP_ID=201&P_FILE_NAME=spa_akt.zip

Wichtige My Oracle Support Note für Patches
My Oracle Support: Real Application Testing Now Available for Earlier Releases (Doc ID 560977.1)

DBA Community Artikel (Inhaltsverzeichnis)
http://tinyurl.com/dbainhalt

Kontaktadresse:

Ulrike Schwinn
Oracle BU DB – Business Unit Database
ORACLE Deutschland B.V. & Co. KG
Riesstr 25, 80992 München
Telefon: +49 89 1430 1865
E-Mail Ulrike.Schwinn@oracle.com
Internet: http://blogs.oracle.com/dbacommunity_deutsch