

Oracle Database 10g: Es muss kein Grid sein

Einmal mehr bietet eine neue Oracle RDBMS Version mehr Flexibilität und einfachere Administration bei insgesamt hochstehender Servicequalität. Da die Verbesserungen sowohl den Administratoren als auch den Entwicklern zu Gute kommen, kann man mit Recht von einer weiteren Reduzierung der Total Cost of Ownership sprechen.

Von Christian Antognini*

Wie bei jeder neuen Software Version gibt es aber Dinge, die man von Anfang an – Out-of-the-box – benutzen kann und andere, die interessant, aber aktuell nur nach intensiven Tests mit gutem Gewissen einsetzbar sind. Einige zentrale Features aus der ersten Kategorie wollen wir uns in diesem Artikel anschauen. In unserem nächsten Artikel (Oracle Database 10g: ein Blick ins Eingemachte) geht es dann um Neuigkeiten aus der zweiten Kategorie, solche also, für die man sich noch etwas Zeit nehmen sollte.

Natürlich können zwei Artikel nicht den gesamten Umfang der Neuigkeiten in Oracle 10g wiedergeben. Wir konzentrieren uns daher auf komplett neue oder zumindest stark erweiterte Aspekte. Für alles weitere – insbesondere auch syntaktische Details – sei hier auf unseren mehrtätigen Kurs zum Thema Oracle 10g verwiesen.

Alle in diesem Artikel vorgestellten Features betreffen die Administration und das Monitoring von Datenbanken – DBA Aufgaben also. Deswegen aber zu meinen, Oracle 10g biete nichts für den Entwickler, wäre falsch, im Gegenteil. Vielmehr hat Oracle im Entwicklungsbereich viele kleinere – aber sehr wichtige – Verbesserungen eingebaut anstatt wenige grosse, neue Features.

Grid Computing und Oracle 10g

Während der letzten Monate hat Oracle das Grid Computing als die nächste Revolution in der Informatik nach dem Aufstieg des Internet bezeichnet. Oracle positioniert dazu die Kombination aus Oracle 10g RDBMS, Application Server 10g und Enterprise Manager 10g (aka Grid Control und Database Control) als die erste komplette Grid Software Infrastruktur und damit Oracle 10g als die Datenbank der Wahl für das Grid.

Unserer Meinung nach ist Oracle 10g aber noch nicht wirklich Grid-aware, zumindest nicht mehr, als es schon Oracle9i ist. Offensichtlich ein hartes Urteil, aber ich denke, sie werden zustimmen, wenn sie den Artikel 'Is Oracle Database Moving Toward Grid Computing?' unter <http://www.trivadis.com> gelesen haben.

Dies ist aber kein wirkliches Problem. Denn nur sehr wenige Firmen benötigen heute schon ein Grid und Oracle 10g hat sehr interessante neue Features zu bieten, die man auch ohne Grid benutzen kann (selbst wenn sie für das Grid entwickelt oder erweitert wurden).

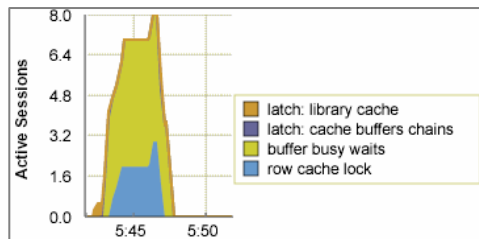
Automatic Workload Repository

Welchem DBA ist es nicht schon einmal passiert, dass ein Anwender anrief und sagte, dass am Vortag Performance Probleme auf der Produktionsdatenbank auftraten und ob der DBA etwas dazu sagen könne... Hier hat der DBA nicht nur ein Performanceproblem am Hals sondern auch ein Informationsproblem: Er wurde viel zu spät informiert! Denn wenn er nicht ein Monitoring Tool mitlaufen liess, kann er kaum mehr tun, als ins Alert Log schauen und evtl. nach Trace Files suchen.

Um Performance relevante Instanzdaten zu sammeln, legt Oracle daher seit Oracle8i Release 2 das Statspack bei. Hat der DBA dieses PL/SQL Package mitsamt einem Job zum Sammeln der Performancedaten installiert, so kann er auch im Nachhinein die problematischsten SQL Statements analysieren, Hit Ratios vergleichen usw. Das Problem von Statspack sind aber sein nicht sehr übersichtlicher Output (ASCII Text only, nicht einmal HTML, von Grafiken ganz zu

schweigen), zudem können immer nur zwei Datenmengen/Zeitpunkte miteinander verglichen, nicht aber eine Sequenz von Daten, d.h. eine Entwicklung eines Wertes, analysiert werden. Dies ist auch genau der Grund, warum Dritthersteller wie die Trivadis mit Tools wie TVD\$STAT hier eingesprungen sind.

In Oracle 10g, wurde das Sammeln von Performancedaten auf eine neue Ebene gehoben. Nicht nur sind die gesammelten Daten in den Data Dictionary der Datenbank integriert, sondern es sind auch per Default Jobs zum Sammeln der Daten aktiviert. Es gibt mit dem Enterprise Manager (Grid Control oder Database Control) ein grafisches Interface und die Betrachtung einer Sequenz von Zeitpunkten ist endlich möglich. Quasi also ein Statspack++. Natürlich musste hier ein neuer Name her, Automatic Workload Repository (AWR) heisst der jüngste Spross in der Familie des Oracle Performance Tunings. Das ‚alte‘ Statspack ist übrigens nach wie vor vorhanden.



Waitinformationen aus dem AWR

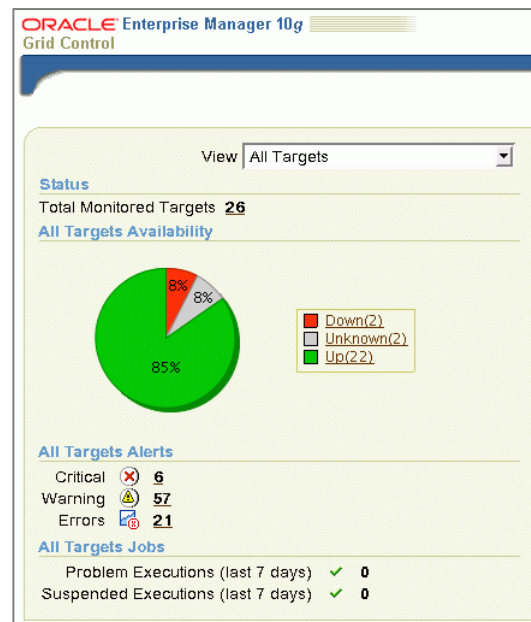
Zusätzlich zu den Snapshots der Performance Daten im AWR kann man die aktuellen Statistiken auch in den bekannten V\$ Views abfragen. Dank dieser Views kann der DBA aktuelle und historische Daten in der gleichen Art und Weise analysieren.

Grid Control

Der neue Enterprise Manager ist nicht nur ein kompletter Rewrite, er hat auch seinen Namen geändert. Nach den Windows Client/Server Tools (OEM 1.x) und den plattform-unabhängigen Java Applikationen (OEM 2.x und 9.x) sind wir nun also bei der dritten Architektur-Generation angelangt: Die ‚alte‘ Infrastruktur mit einem Java Management Server wurde durch das Grid Control ersetzt, die Standalone Console (aka DBA Studio) durch das Database Control.

Beiden ist gemeinsam, dass sie als Java Server Pages mit einem Applikationsserver arbeiten und komplett über einen Browser bedienbar sind. Und dies als ‚echte‘ Web Applikation, nicht einfach ein n-hundert Megabyte Applet das von einem zentralen Server geladen wird.

Database Control arbeitet dabei mit genau einer 10g Datenbank, Grid Control bedient mehrere Systeme und kommt auch mit Oracle8i and Oracle9i Datenbanken klar.



Die Abbildung zeigt einen Teil der Einstiegsseite des OEM, welche dem DBA eine Übersicht über den Zustand all seiner Systeme gibt.

Transportable Tablespaces

Transportable Tablespaces wurden in Oracle8i eingeführt, um den Transfer von grossen Datenmengen zwischen Datenbanken zu beschleunigen. Hier muss man nur noch Metadaten zu den transportierten Tablespaces exportieren und importieren, die Daten selber werden durch OS-Level Copy der Datenfiles an ihr Ziel gebracht. Das Einklinken der Tablespaces in die Zieldatenbank geschieht beim Importieren der Metadaten. Der Hauptvorteil dieses Ansatzes ist die Vermeidung der Resource intensiven Daten-Export/Import Operationen.

Da beim Transportieren eine 1 zu 1 Kopie der Datenfiles gemacht wird, können nur Tablespaces transportiert werden, deren Datenfiles mit den Datenfiles der Zieldatenbank kompatibel sind. Was heisst nun kompatibel? In Oracle8i bedeutete dies, dass die Datenbanken auf dem gleichen Betriebssystem laufen, die selbe Datenbankblockgrösse und die selben Zeichensätze benutzen (Beachten sie die Mehrzahl: Sowohl Character Set als auch National Character Set müssen gleich sein!). Das sind ziemlich strenge Voraussetzungen, gerade bei der Königsdisziplin ‚Datentransport aus dem OLTP System ins Datawarehouse‘ wird die Bedingung der gleichen Datenbank Blockgrösse meistens verletzt sein.

Glücklicherweise hat Oracle schon in Oracle9i mit dem Multi-Blocksize Support einen wesentlichen Stolperstein aus dem Weg geräumt. Der Datentransport aus dem OLTP System ins Warehouse geht damit problemlos, die Zieldatenbank muss einfach einen Buffer Cache mit der Blockgrösse der Quelldatenbank konfiguriert haben.

Der grosse Durchbruch kam aber mit Oracle 10g: Der Transport von Tablespaces ist nun auch über Betriebssystemgrenzen hinweg möglich. Prinzipiell läuft der Vorgang gleich ab wie seit Oracle8i, nur wenn die CPUs der beteiligten Betriebssysteme ein unterschiedliches Byte-Ordering verwenden, muss eine Konversion der Datenfiles mittels RMAN durchgeführt werden (und dauert damit in etwa so lange, wie eine RMAN Backup des Datenfiles dauern würde).

Das TTS Feature wird von Oracle als eines der neuen Grid Features positioniert: Daten sind damit zwischen den Knoten eines Grids verschiebbar. So sehr man sich über diese Positionierung streiten kann, das Feature an sich ist genial und ermöglicht uns die Plattform Migration auch bei Datenmengen, bei denen Export/Import nicht mehr möglich ist (zeitlich oder mengenmässig). Stellen Sie sich eine Oracle Datenbank auf einem Network Attached Storage System (NAS) vor: Die Migration von einer Sparc Solaris auf eine Intel Linux Maschine wäre nur noch ein Export/Import der Metadaten plus RMAN Durchlauf, da die Datenfiles an sich durch das NFS Protokoll ohne Filesystem Konversion auf beiden Maschinen sichtbar gemacht werden können!

Recycle Bin

Jeder von uns hat schon einmal aus Versehen die falsche Tabelle gedropped. Geben Sie es zu! Vor Oracle 10g war die einzige Rettung für normal Sterbliche (solche ohne Zugang zu Oracle internen Tools) eine Database Point in Time Recovery, zumindest aber eine Tablespace Point in Time Recovery (je nach Oracle Version war letzteres keine Freude...). Glücklicherweise, wer in solch einer Situation ein einigermaßen aktuelles logisches Backup (Export) der Tabelle hatte.

Mit Oracle 10g gibt es eine viel einfachere Lösung:

```
FLASHBACK TABLE table name TO BEFORE DROP
```

So einfach? Ja, so einfach! Der Grund liegt darin, dass Tabellen nicht mehr gedropped, sondern (mitsamt Constraints, Triggern und Indizes) nur umbenannt werden. Sie wandern in einen logischen Papierkorb, aus dem sie zurückgeholt werden können, sofern der Platz nicht wieder verwendet wurde (dies geschieht aber nur, falls eine Data Management Operation keinen anderen freien Platz mehr gefunden hat).

Backup und Recovery

Seit Oracle8 empfiehlt Oracle das Backup mit dem Recovery Manager (RMAN). Die meisten Backup/Recovery Neuerungen betreffen seitdem auch dieses Tool. In Oracle 10g fristen die RMAN Neuerung marketingmässig das Dasein einer grauen Maus. Äusserst bedauerlich, der Betriebs-DBA wird ob der Neuerung in Verzückung geraten!

So zum Beispiel die inkrementellen Backups. Diese waren bis jetzt ja nicht schneller als die Full Backups, meistens sogar langsamer. Dies lag daran, dass Oracle die gesamte Datenbank nach geänderten Blöcken scannen musste. Der Gewinn der inkrementellen Backups lag daher nur in der geringeren Grösse. Ab Oracle 10g kann Oracle eine Bitmap verwalten, die die geänderten Blöcke kennzeichnet. Auf diese Art sind Performancegewinne im Bereich von Faktoren möglich, die Backupzeit ist proportional zur Anzahl der geänderten Blöcke!

Grundsätzlich sind Backups neu auch komprimierbar, man muss als nicht mehr manuell die Backup Pieces nach dem Backup durch COMPRESS oder GZIP schleusen.

Bis Oracle9i einschliesslich war es empfohlen, nach einer Database Point in Time Recovery ein Backup zu erstellen. Durch das OPEN RESETLOGS konnte ein Backup, welches VOR dem OPEN erstellt wurde, nicht mehr für eine Recovery NACH dem OPEN verwendet werden. Es gab ab Oracle9i zwar Anleitungen, wie man sich hier helfen konnte, aber ohne Übung (z.B. in unserem Advanced Backup Recovery Workshop ☺) war das nicht wirklich rastsam. Ab Oracle 10g existiert dieses Problem nicht mehr. Ohne irgendeine Konfiguration, es ist einfach weg. Oracle läuft bei einer Recovery über ein OPEN RESETLOGS hinweg.

Überhaupt wurde der Recovery Prozess in vielen Bereich schlauer. Existierten z.B. zwar alle archivierten Redo Logs seit dem Erzeugen eines Datafiles, nicht aber ein Backup des Datafiles, so musste man eine leere Hülle für das Datafile mit ALTER DATABASE CREATE DATAFILE erstellen, um die Recovery fortsetzen zu können. Macht Oracle nun selber. Oder ein Backup Piece, welches korrupt war: Musste man manuell auf UNAVAILABLE setzen. Oracle nimmt heute selbstständig ein älteres Piece. Tablespace Point in Time Recovery? Früher nur durch manuelles Erstellen einer Auxiliary Instanz machbar. Heute macht Oracle das selber, eine TSPITR ist ein Einzeiler geworden.

Resumé und Ausblick

Wir denken, dass man sich zu Recht über diverse Features in Oracle 10g freuen darf. Ohne Grid und ohne Exabyte-grosse Datenbanken bringt es auch der ‚kleinen‘ aber deswegen nicht weniger wichtigen Kundenstammverwaltung wertvolle Erleichterungen.

Im zweiten Teil des Artikels wollen wir uns dann auf etwas exotischere Neuerungen konzentrieren. Features wie Automatic Storage Manager und Data Pump mögen heute noch nicht im allgemeinen Gebrauch sein, sie werden aber in ein paar Jahren sicherlich zu den Standardwerkzeugen des DBAs zählen.

* Christian Antognini ist Senior Consultant und Trainer mit Spezialgebiet Oracle Database Server bei der Trivadis AG in Zürich, Schweiz. Weitere Infos unter www.trivadis.com