Oracle 10*g*: ein Blick ins Eingemachte

Christian Antognini und Dr. Martin Wunderli*

Dies ist der letzte Teil des zweiteiligen Artikels über neue Features in Oracle 10*g*. Der erste Teil, Oracle Database 10*g*: Es muss kein Grid sein, hat diejenigen zentralen, neuen Features behandelt, welche man ohne grösseren Aufwand in die Produktion bringen kann. In diesem Artikel wollen wir uns mit Dingen befassen, die zwar äusserst interessant sind, aber etwas mehr Einarbeitungs-, Test- und Planungsaufwand erfordern, um den grösstmöglichen Nutzen zu erreichen und die für den Betrieb notwendige Stabilität zu erhalten. Schauen wir also, was Automated Storage Management, Cluster Ready Services, Flashback Database und die Data Pump uns bringen werden.

Automated Storage Management

Die meisten Daten, die von einer Datenbank verarbeitet werden, werden schlussendlich in einem Storage Subsystem abgelegt. Jedes Betriebssystem stellt hierzu spezielle Verwaltungs- und Zugriffskonzepte zur Verfügung. Die direkteste (und älteste) Variante ist der direkte, ungepufferte Zugriff, auch bekannt als Raw Devices. Zwischen einer Applikation (hier dem Oracle RDBMS) und einem RAW Device befinden sich keine OS Schichten wie z.B. zusätzliche Caches oder Filesystem Logik, sodass die I/O Performance nahezu optimal ist (Zugegeben, die Welt ist nicht mehr so simpel wie beschrieben. Dafür sorgen u.a. Dinge wie logische Volume Manager im OS oder auf externen Disksubsystemen. Aber vom Prinzip her stimmt es.).

Das Fehlen einer Schicht zwischen Applikation und Hardware hat aber auch Nachteile, der wichtigste ist der Mangel an Flexibilität. Aus diesem Grunde gibt es Filesysteme, sie abstrahieren von der physischen Schicht der Devices und ermöglichen der Applikation eine sehr viel flexiblere Handhabung der Speichersysteme. Es ist natürlich unvermeidbar, dass diese zusätzliche Schicht auch Overhead mit sich bringt, insbesondere, weil sie sie die Anforderungen vieler unterschiedlicher Applikationen erfüllen müssen. Aus diesem Grunde haben Hersteller wie Veritas in Zusammenarbeit mit Oracle neue Filesysteme entwickelt, die die speziellen Bedürfnisse eines Datenbanksystems berücksichtigen und so die Flexibilität eines klassischen Filesystems (Fileallokation) mit den Performance Vorteilen von RAW Devices verbinden. Nachteil ist aber, dass hier ein weiterer Hersteller ins Boot kommt und die Filesysteme relativ teuer sind.

Um die genannten Nachteile der Spezialfilesysteme unter Beibehaltung der Vorteile zu eliminieren und um die Gesamtkosten eines Datenbankservers zu senken, hat Oracle nun ein eigene Schicht des Storage Managements entwickelt, welche die Funktionialität von Volume Manager und Filesystem vereint und speziell auf die Bedürfnisse einer Oracle Datenbank – insbesondere im Grid Umfeld – zugeschnitten ist: Automated Storage Management (ASM).

Die Verantwortung für Storage verschiebt sich hierbei vom Systemadministrator zum DBA und zur Datenbank. Während früher der Systemadministrator Disks zu Mirrors und Stripesets zusammengefasst und darauf Filesysteme erstellt hat, ist er heute nur noch für das Ändernn des Besitzers des RAW Devices zuständig. Der DBA gruppiert die Devices dann in Gruppen (mit 0, 1 oder 2 Spiegeln) und erstellt darauf Tablespaces. Dinge wie Striping werden aber vom Datenbanksystem automatisch erledigt, weder Systemadminsitrator noch DBA müssen sich darum kümmern. Auch um eine Verzeichnishierarchie muss der Systemadministrator oder der DBA nicht mehr besorgt sein (der DBA könnte es zwar, aber wir empfehlen es nicht). Der DBA definiert einfach die Diskgruppe, auf der ein Tablespace liegt, dieser wird dann von Oracle automatisch plaziert und verteilt und die entsprechende Ordnerstruktur wird erstellt.

```
CREATE TABLESPACE users DATAFILE '+DISK_GROUP_01' SIZE 200M;
```

Beachten Sie, dass eine Diskgruppe durch das vorangestellte '+' gekennzeichnet wird.

Abbildung 1: Die Diskgruppe DISK_GROUP_01 besteht aus zwei sogenannten Failure Groups (aka Mirrors oder Spiegel).

Failure Group	Path
DISK_GROUP_01	
DISK_SET_01	
ASM_DISK_02	/dev/raw/raw2
ASM_DISK_01	/dev/raw/raw1
DISK_SET_02	
ASM_DISK_04	/dev/raw/raw4
ASM_DISK_03	/dev/raw/raw3

Jede Failure Group besteht aus zwei Disks. Über die Disks einer Failure Group wird immer gestriped, je nach Filetyp mit unterschiedlichen Stripegrössen. Vom Prinzip her also wie Raid 0+1. Das Interessante ist nun, dass beim Hinzufügen von Disks in eine Failure Group, auch die existierenden (!) Tablespaces umverteilt werden, sodass sie über die neue Disk gestriped sind! Zudem: Da die Disks in einer Diskgroup schlussendlich RAW Devices des Betriebssystems sind, werden Datenbankblöcke, Redo Logs, Backup Pieces usw. mit der Performance von RAW Devices geschrieben.

Um mit ASM zu arbeiten, braucht es eine Menge von Prozessen, welche die administrativen Arbeiten (z.B. das Rebalancing) erledigt. Eine Menge von Prozessen ist im Oracle Jargon eine Instanz, per default heisst sie +ASM. Wohlgemerkt, eine Instanz, keine Datenbank. Also keine Datenfiles, keine Controlfiles, nur ein Parameterfile, welches zudem minimalst ist: Der eigentliche Aufbau der Diskgroupen ist auf diesen selbst gespeichert. Es ist zudem wichtig zu verstehen, dass I/O immer direkt von der Datenbankinstanz zu den Disks geht, nie über eine ASM Instanz!

Quasi als 'Verzeichnisdienst der ASM Instanzen', der auch im Clusterumfeld zum Tragen kommt, dient der Oracle Cluster Synchronization Service Daemon (OCSSD), der bei der Installation der Oracle Software schon fest im System verankert wurde (Andersherum: Wenn Sie kein ASM brauchen, brauchen Sie auch diesen Daemon nicht). Abbildung 2 zeigt die verschiedenen Komponenten.

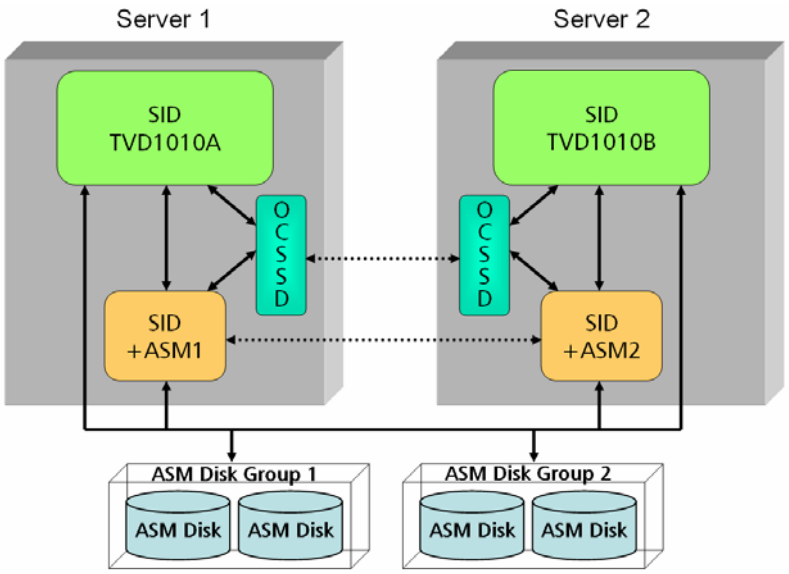


Abbildung 2: Clustered ASM Konfiguration.

Wenn wir uns die Anforderungen an Datenbank Storage anschauen – einfache Adminsitration, sehr gute Performance bei keinen zusätzlichen Kosten – dann sieht man, dass ASM sie alle erfüllt. Die Administration ist zentralisert, kann über Kommandozeile (SQLPLUS) oder über GUI (Database oder Grid Control) erfolgen und ist soweit wie möglich automatisiert. Natürlich wird jeder DBA ein so zentrales Features zuerst einmal in seiner Umgebung intensiv testen, insbesondere das Verhalten bei Diskausfällen, aber man kann jetzt schon sagen, dass Oracle mit ASM ein grosser Wurf gelungen ist.

Real Application Clusters

Über die letzten Jahre hat sich Real Applications Clusters – der ehemalige Oracle Parallel Server – zu einer zuverlässigen und performanten Cluster Lösung gemausert: Sicherlich eine der interessantesten (wenn auch eine der teuersten) Optionen der Oracle Enterprise Edition. In Oracle 10*g* wurde die Feature Liste der RAC Option nicht wesentlich verändert, es gibt aber drei Änderungen, die die Kosten von RAC massiv senken und damit auch seine Verbreitung erhöhen werden.

Die erste ist, dass Oracle ab sofort auf allen Plattformen, auf denen RAC erhältlich ist, eine eigene Cluster Manager Software mitliefert: Cluster Ready Services (CRS). Schon in Oracle 9*i* traf dies auf die Windows und Linux Plattform zu, jetzt gilt es auch für Solaris, HP/UX, etc. Die Kosten für ein Dritthersteller Produkt fallen damit weg, CRS ist in die RAC Option integriert. Will man aus Gründen der Einheitlichkeit oder weil man dem neuen Produkt noch nicht so traut, weiter mit einer anderen Cluster Software arbeiten, ist dies aber weiterhin möglich. Oracle positioniert CRS als zentrale Komponente der Datenbank Schicht im Grid Computing. Mit der entsprechenden Marketingbrille kann man das so sehen, freuen wir uns vorerst einfach über die reduzierten Kosten und die geringere Komplexität eines RAC Systems.

Die zweite Verbesserung, die für RAC relevant ist, ist ASM. Während im Single Instance Betrieb vielleicht einige 'nur' wegen Striping und RAW Device Performance nicht so den Drang zum Wechsel der Storage Architektur verspüren, ist das bei RAC anders. Hat man kein spezielles Cluster Filesystem oder Network Attached Storage (beides teuer), dann ***muss*** man hier mit RAW Devices arbeiten, Dinge wir Backup der archivierten Redo Logs sind nicht mehr so trivial. Mit ASM ist das kein Thema mehr: Datenfiles, Controlfiles, archivierte Redo Logs, Server Parameter Files etc. sind mit exzellenter Performance von allen Knoten mit einfachen Mitteln handhabbar.

Die dritte Verbesserung ist nicht mehr technischer Natur: Die RAC Option ist in der Standard Edition kostenfrei dabei, falls der Cluster total nicht mehr als 4 Knoten hat. D.h. ich kann heute eine (relativ) kleine Datenbank, z.B. der Katalog eines Internetshops, anstatt auf einem 4-Prozessor Intel Server zu gleichen Oracle Lizenzkosten auf zwei 2-Prozessor Knoten hochverfügbar machen! Dass hier kein Cluster Manager eines Drittherstellers unterstützt ist, kann man wohl verschmerzen.

Flashback Database

Im Fall einer logischen Datenkorruption oder Benutzerfehlern war es der DBA gewohnt, eine Point-In-Time Recovery des Tablespaces, meistens aber der Datenbank durchzuführen. Dieser Vorgang erforderte zwei Schritte: Das Zurückholen der Datenfiles sowie das forwärts Rollen der Datenfiles mit Hilfe der inkrementellen Backups oder archivierten Redo Logs. Technisch funktioniert das gut, bei einem 2 TByte Data Warehouse überlegt man sich diesen Schritt aber zweimal. Ein versehentliches Drop Table wird dann vielleicht lieber auf eine andere, aufwändigere Art – aber eben doch schneller als bei einem Full Database Restore – erledigt.

Ein bessere Ansatz wäre es, die Datenbank mit Hilfe der Online und archivierten Redo Logs zurückzurollen. Dies kann Oracle aber nicht, denken wir nur an Kommandos wie Truncate Table. Zurückrollen an sich ist aber schon gut, und das hat Oracle auch mit einem neuen Feature implementiert, allerdings nicht auf Basis der Redo Logs.

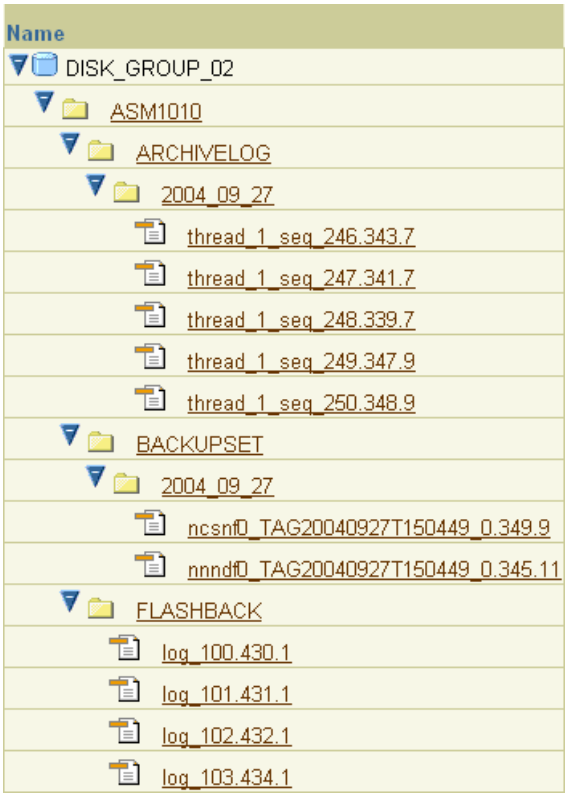
Fall die Datenbank im sogenannten Flashback Modus betrieben wird (ALTER DATABASE FLASHBACK ON im Mount Status), kopiert ein neuer Hintergrundprozess (RVWR) in regelmässigen Zeitabständen die Before Images von geänderten Blöcken in einen speziellen Bereich auf Disk, der Flash Recovery Area. Will man nun die Datenbank auf einen vergangenen Zeitpunkt zurücksetzen, so werden die entsprechend älteren Block Images der geänderten Blöcke aus der Flash Recovery Area zurückgeholt und mit Hilfe der Redo Logs auf den

gewünschten Zeitpunkt recovered (die Datenbank muss also im ARCHIVELOG Modus betrieben werden). Wie weit man mit dieser Methode zurückgehen kann, wird über einen Instanzparameter (DB_FLASHBACK_RETENTION_TARGET) bestimmt.

Es ist zu beachten, dass die mit FLASHBACK DATABASE zurückzusetzenden Datenfiles vorhanden und auch nicht durch Media Fehler korrumpiert sein dürfen. Die Flash Recovery Area ist daher kein Ersatz für ein normales Backup!

Abbildung 3: Flash Recovery Area

Die Flash Recovery Area ist eine von Oracle verwaltete Verzeichnisstruktur auf dem Filesystem oder in einer ASM Diskgruppe. Neben den Flashback Logs können darin (müssen aber nicht) alle Backup & Recovery relevanten Dateien wie Backup Pieces, archivierte Redo Logs, usw. abgelegt werden. Die Struktur ist unter kompletter Oracle Kontrolle, der DBA muss nur die Gesamtgrösse definieren. In Abbildung 3 ist eine Flash Recovery Area gezeigt, die auf einer ASM Diskgruppe plaziert ist. Sie enthält neben den beschriebenen Flashback Logs auch Backup Sets/Pieces und archivierte redo Logs.



Data Pump

Jeder DBA kennt die Utilities zum logischen Sichern und Zurückholen von Daten: EXP und IMP. Über die letzten 10 Jahre hat Oracle nur wenige wirklich neue Features in diese eingebaut, die Architektur blieb sogar komplett die gleiche. Wenn wir auf die Zeit zurückblicken, als EXP und IMP entwickelt wurden, erkennen wir, dass die Datenbanken damals viel schneller und einfacher aufgebaut waren (XML war noch ein Fremdwort, XDB erst recht). Es ist also nicht verwunderlich, dass Oracle entschieden hat, die beiden Tools komplett zu ersetzen: Data Pump Export (EXPDP) und Data Pump Import (IMPDP). Die 'alten' Tools sind vorerst aber immer noch vorhanden und auch unterstützt.

Das Hauptziel der neuen Architektur ist die Geschwindigkeitsteigerung beim Export und Import. Dies wird vor allem – aber nicht nur – erreicht durch Parallelisierung. Genauer: Für eine bestimmte Operation sind jeweils mehrere Slave Prozesse und Datenfiles unterstützt. Andere Features machen Export und Import flexibler, z.B. Job Management, dynamisches Hinzufügen von Dump Files und Prozessen zur Laufzeit und Import/Export über Database Links!

Die wichtigste Änderung in der Architektur besteht darin, dass das ganze Processing durch das Package DBMS_DATAPUMP abgewickelt wird. Wir können wir also von einem Server Managed Export sprechen, so wie wir von einem Server Managed Backup sprechen. EXPDP und IMPDP sind dabei nur noch Steuerprogramme wie RMAN ein reines Steuerprogramm für den eigentlich

Backup Server Prozess ist. Ein willkommenere Nebeneffekt der Package basierten Architekutr ist, dass man aus einer PL/SQL Applikation ein Export oder Import starten kann ohne ein externes Programm aufrufen zu müssen.

Abbildung 4 zeigt die Hauptkomponenten der neuen Architektur. Das expdp/impdp Client Programm wird nur zum Starten/Stoppen/Wideraufnehmen des Processings oder zum dynamischen Ändern von Parametern benötigt. Genauer: Ist der Export Prozess einmal gestartet, kann der Client beenden und alles im Hintergrund ablaufen lassen. Denn die eigentlich Kontrolle wird durch den Master Control Prozess übernommen, er schreibt auch die Logs und managed eine sogenannte Master Table welche Metainformationen über das Dump File enthält. Man kann sich die Master Table als das Data Dictionary des Dump File Sets vorstellen. Die eigentliche Arbeit wird aber von sogenannten Worker Prozessen oder von deren Slave Prozessen durchgeführt. Kommunikation erfolgt jeweils über Queues, also asynchron.

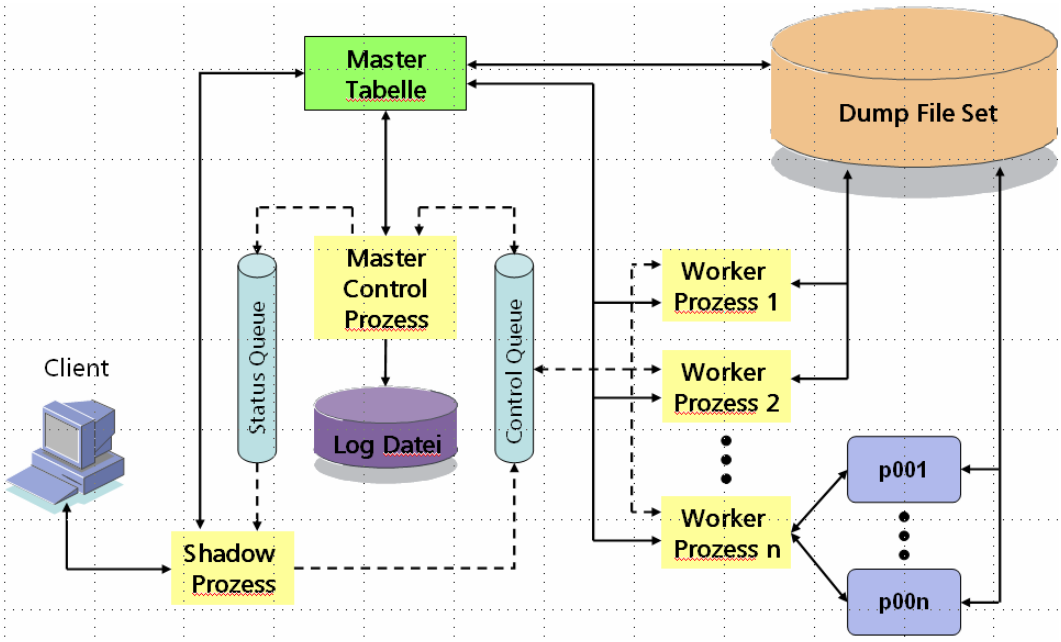


Figure 3: Die Architektur der 'Datenpumpe'

Data Pump wurde als spezieller Treiber (neben dem bisherigen ORACLE_LOADER) auch in die externen Tabllen integriert. Damit ist es möglich, externe Tabellen mittels CREATE TABLE aus Oracle heraus zu erstellen. Leider ist das externe Tabellen Format dann aber ein binäres und kann nur mittels dem DATA_PUMP Treiber wieder gelesen werden (d.h. nicht mit Oracle9*i*), ein echtes SQLUNLDR als Pendant zum SQLLDR fehlt also immer noch.

Fazit

Es ist offensichtlich, dass Oracle mit der Version 10*g* nicht nur das Datenbank Management vereinfacht, sondern auch Aufgaben (und damit Business) von Partnern wie EMC und Veritas übernehmen möchte. Sollten sich ASM und CRS langfrisitg als genügend stabil und unterstützt erweisen, werden diese Hersteller mit mehr als nur I/O Balancierung, einem Filesystem oder einem Cluster Manager die Oracle Kunden überzeugen müssen. Uns Kunden kann das nur Recht sein, Konkurrenz belebt das Geschäft! Auch ist klar, dass bei dem durch Oracle massiv unterstützten Trend in Richtung Linux die Kostenersparnis nicht nur beim Betriebssystem liegen kann.

Oracle 10*g* hat also für jeden etwas dabei. Für den auf Stabilität bedachten DBA, der einfach das Bewährte besser haben möchte, so zum Beispiel ein grafisches Statspack in Form des

Automated Workload Repository oder schnellere RMAN Backups durch Block Change Tracking. Oder für den DBA, der zu neuen Ufern aufbrechen will, und mit Cluster Ready Servcies die Anzahl Hersteller in einer Hochverfügbarkeitslösung reduziert oder indem er mit Automatic Storage Management in eine Domäne einbricht, die bis jetzt den Systemadministratoren vorbehalten war.

Für beide gilt aber, dass ihre Arbeit auch mit Oracle 10*g* alles andere als überflüssig wurde. Der DBA kann aber schwierigere Aufgaben einfacher und schneller erledigen und ist damit für die Herausforderungen zukünftiger Applikationen besser gewappnet. Wer sonst noch Argumente für den DBA Job braucht, suche einfach nach 'dba job future' auf asktom.oracle.com ☺

Und der Entwickler? Wurde er mit Oracle 10*g* etwa vergessen? Mitnichten! Nur, wie schon bei Oracle 9*i* scheint der erste Wurf eines neuen Major Releases tatsächlich mehr DBA lastig zu sein. Zumindest was die 'grossen Knaller' betrifft. Wichtige Verbesserungen gibt es aber auch für die Entwickler, so viele, dass wir einen 2-tägigen 'Oracle 10*g* – New Features für Entwickler' Kurs anbieten! Und zudem spielt die Musik für den Entwickler immer mehr im Bereich der Application Server. Und hier hat Oracle mit dem IAS 10*g* wirklich einiges gegenüber IAS 9*i* getan...

Gleichgültig ob Entwickler oder DBA, falls Sie Schulung, Consulting oder Coaching brauchen, sind Sie bei der Trivadis an der richtigen Adresse. Seit dem 1.10. auch mit unserer 10. Niederlassung (Hamburg) und im 10-ten Jahr seit der Gründung. Ach ja, unbestätigten Gerüchten zufolge hat Oracle die Version 10 solange verzögert, damit der neue Release passend zu unserem Jubiläum kommt... ☺

* Christian Antognini und Dr. Martin Wunderli sind Senior Consultants und Trainer bei der Trivadis AG in Zürich-Glattbrugg. Ihre langjährige Erfahrung von der Architektur bis zur Implementierung Oracle RDBMS basierter Systeme führt sie zu Kunden in der Schweiz und Deutschland. Mehr Informationen zur Trivadis finden Sie auf http://www.trivadis.com