

Datenbank-Virtualisierung & Instance Cloning und was macht Oracle 12c?

Sebastian Winkler
CarajanDB GmbH
Erfststadt

Schlüsselworte

Database Cloning, Database Duplication, Oracle 11g, Oracle 12c, Direct NFS, Copy-On-Write

Einleitung

Wenn das Wort „Clone“ fällt, denkt man im ersten Moment sicherlich an die Clone-Krieger aus Star Wars. Dabei stand sicherlich im Vordergrund, dass das Original und die Clones möglichst identische Funktionen aufwiesen, d.h. gleich stark, schnell und was sonst noch sein sollte.

Wenn es um das Cloning von Datenbanken geht, verhält es sich nicht anders. Ziel ist es, eine möglichst identische Kopie der Produktionsumgebung zu erstellen, um darauf zu testen oder zu entwickeln. Meist bleibt es dann nicht bei einer einzelnen Kopie, sondern für unterschiedliche Anforderungen müssen verschiedene Kopien zur Verfügung gestellt werden.

In diesem Vortrag werden die verschiedenen Verfahren kurz beschrieben um dann exemplarisch zwei Lösungen, CloneDB und Delphix näher zu beleuchten.

Möglichkeiten von Cloning

Die Notwendigkeit Datenbanken für Tests oder Entwicklungen zu kopieren, gibt es sicherlich schon genau so lange, wie es Datenbanken gibt. Dabei wird in der Regel ein Backup der Datenbank gemacht und dieses dann auf einem anderen Server recovered. Die dafür benötigte Zeit und der Platzbedarf sind dabei in der Regel die limitierenden Faktoren, d.h. es müssen Kompromisse hinsichtlich Aktualität und Anzahl gemacht werden.

Seit Oracle 11g gibt es eine ganz simple Methode, eine Datenbank mit dem RMAN zu klonen. Dafür meldet man sich an die Quelldatenbank (TARGET) und eine Instanz auf dem Zielsystem (AUXILIARY) an und nutzt den DUPLICATE Befehl für die Erstellung der Kopie.

```
% rman target sys/manager@PROD auxiliary sys/manager@MYCLONE
```

```
RMAN> DUPLICATE TARGET DATABASE;
```

Dabei wird auf ein bereits erstelltes Backup der primären Datenbank zurückgegriffen und mit Hilfe der archivierten Redolog-Dateien kann die Clone-Datenbank dann auf jeden beliebigen Stand vorgeollt werden.

Seit Oracle 11g Release 2 ist sogar ein direktes Clonen, d.h. ohne vorheriges Backup der primären Datenbank, möglich. Der Befehl ändert sich dabei nur unwesentlich:

```
RMAN> DUPLICATE TARGET DATABASE FROM ACTIVE DATABASE;
```

Natürlich gibt es noch eine Reihe von Optionen, wie z.B. eine Angabe der Verzeichnisse für die Datenbank-Dateien, aber in der Regel reicht dieser Befehl tatsächlich aus, um einen Clone zu erstellen.

Mit Oracle 12c und den vielen tollen Funktionen der Multitanant (Pluggable) Database ist es jetzt sogar noch einfacher, einen Datenbank Clone zu erstellen. Man kann entweder ein eigenes Pluggable Database Template hinterlegen oder auch direkt aus einer existierenden Pluggable Database eine Kopie erstellen. Allerdings gibt es hier „noch“ ein paar Wehrmutstropfen: Zunächst einmal muss die zu kopierende PDB „Read-Only“ geöffnet sein. Für ein schnelles Kopieren einer Datenbank reicht dies in der Regel nicht aus – aber warten wir’s ab, irgendwann kommt sicherlich Oracle 12c Release 2 (oder ein kleines Patch).

Was ist also das Problem?

Für einen Clone reicht dieses Verfahren sicherlich aus, doch was ist, wenn mehrere Clones erstellt werden müssen. Das Backup belastet die Produktion, d.h. es muss mit Performancebeeinträchtigungen gerechnet werden. Außerdem ist die Laufzeit natürlich auch nicht unerheblich. Eine Datenbank, die mehrere Terabyte groß ist, wird man nicht in einer Stunde kopiert haben. Viel wichtiger ist aber, dass jeder Clone den gleichen Platz wie die Produktion benötigt. Da wird der Anzahl der erstellten Clones schnell eine Grenze gesetzt.

Viele Unternehmen behelfen sich an dieser Stelle mit dem Data Pump. Anstelle der gesamten Datenbank wird nur noch der Teil, der gerade für den Test oder die Entwicklung notwendig ist, kopiert. Das spart zwar Plattenplatz, ist in der Regel jedoch langsamer, weil zunächst eine Rumpfdatenbank aufgebaut sein muss und, wie wir wissen, ist DataPump nicht gerade die schnellste Möglichkeit zum Kopieren großer Datenmengen. Auch wenn, ähnlich wie beim DUPLICATE mit dem Parameter NETWORK_LINK ohne Zwischenspeicherung von Dump-Files gearbeitet werden kann, ist auch diese Methode Zeit- und Platzintensiv.

Vielleicht kann man ja von der Virtualisierung lernen.

In virtuellen Umgebungen haben sich mittlerweile Verfahren etabliert, bei denen es eine lesende Version und eine geänderte Version eines Datums gibt. D.h. mehrere Guest Systeme greifen bei lesenden Zugriffen auf die gleiche Ressource zu und nur die Änderungen werden in einer eigenen Kopie gespeichert. Dieses sogenannte Copy-On-Write (COW) wird unter anderem bei ZFS Storage verwendet. Der Vorteil liegt auf der Hand: es wird weniger Plattenplatz verbraucht und der Clone kann wesentlich schneller zur Verfügung gestellt werden, denn es müssen ja zunächst nur Zeiger auf die bereits existierenden Dateien zur Verfügung gestellt werden.

CloneDB

Oracle hat mit der Version 11.2.0.2 (also mit dem Patchset!) still und heimlich eine neue Möglichkeit für das Duplizieren von Datenbanken zur Verfügung gestellt. CloneDB erlaubt es, auf Basis einer RMAN Image Kopie beliebig viele Datenbanken aufzusetzen. Grundlage hierfür ist die bereits mit Version 11.2.0.1 eingeführte Direct NFS Unterstützung in der Datenbank. Mit Direct NFS ist es zunächst möglich, dass eine Oracle Datenbank über einen speziellen Treiber (dNFS) auf NFS-Laufwerken betrieben werden kann. Mit CloneDB wird jetzt eine zusätzliche I/O Schicht eingeführt, so dass lesende Zugriffe aus dem RMAN Backup bedient werden, während schreibende Zugriffe in eigene Dateien erfolgen. In der Datenbank selbst sieht es allerdings so aus, als ob ganz normale Datafiles benutzt werden, d.h. ein Administrator bzw. eine Anwendung kann nicht erkennen, dass hier ein Clone benutzt wird.

CloneDB lässt sich, wie man im Vortrag sehen wird, ganz leicht aufsetzen und die Erstellung eines weiteren Clones geschieht in Minuten und verbraucht so gut wie keinen Platz, egal wie groß die Quelldatenbank ist.

Nachteil von CloneDB ist allerdings, dass alle Kopien auf einer Image Kopie aufsetzen, d.h. je älter die Image Kopie umso mehr „lokaler“ Speicher wird benötigt.

Delphix

Hier kommt die Firma Delphix (www.delphix.com) ins Spiel. Über eine virtuelle Appliance werden Quelldatenbanken verwaltet und, ähnlich wie bei CloneDB, wird dann ein RMAN Backup gemacht. Aber im Gegensatz zum CloneDB wird dieses Backup komprimiert, d.h. es belegt wesentlich weniger Platz und zu festgelegten Zeiten werden außerdem inkrementelle RMAN Backups hinzugefügt, so dass ein Clone auf jedem beliebigen Punkt aufgebaut werden kann.

Eine fancy Web-Oberfläche vervollständigt die Anwendung, so dass es auch ohne DBA Knowhow möglich wird, sich einen Clone zu erstellen, diesen zu aktualisieren oder wiederum Clones auf Clones aufzubauen. Und falls ein Clone länger benötigt wird, kann daraus auch wieder eine reguläre Datenbank aufgebaut werden.

Im Unterschied zur Lösung von Oracle kommt hier eine proprietäre Filesystem-Lösung (DxFS) zum Einsatz. Allerdings merkt man davon überhaupt nichts, da auch der Direct NFS Treiber für die Oracle Datenbank nicht notwendig ist, sondern die Voraussetzung für die Nutzung ausschließlich eine identische Oracle Software Installation ist. Auch Listener oder vordefiniertes SPFILE sind nicht notwendig, alle Dateien werden über Delphix verwaltet.

Fazit

Da Oracle Datenbanken die unangenehme Eigenschaft haben, dass sie immer größer werden, sollten wir uns jetzt mit Copy-On-Write Technologien auseinandersetzen. CloneDB ist sicherlich eine Lösung, die man sich näher ansehen sollte, auch wenn der Aufwand derzeit nur lohnt, wenn mehrere Clones erstellt werden müssen. Delphix liefert eine fertige Lösung, die mit wenig Aufwand implementiert werden kann, allerdings mit dem Nachteil recht hoher Kosten für das Produkt selbst.

Kontaktadresse:

Sebastian Winkler
Carajandb GmbH
Siemensstraße 25
D-50374 Erftstadt
Telefon: +49 (22 35) – 1 70 91 88
Fax: +49 (22 35) – 1 70 89 79
Mobil: +49 (1 75) – 8 64 90 61
E-Mail: sebastian.winkler@carajandb.com
Internet: www.carajandb.com