Multitenant Database – DIE Lösung für DaaS?

Johannes Ahrends
CarajanDB GmbH
Erftstadt

Schlüsselworte
Oracle 12c Multitenant Database, Database As A Service, Real Application Clusters, Data Guard

Einleitung

Die Mandantenfähigkeit der Datenbank mit Version 12c ist für mich die größte Änderung in der Oracle Architektur seit der Einführung von Oracle Parallel Server mit der Version 6.2 im Jahr 1991. Von Mitbewerbern teilweise belächelt, weil sie meinen, etwas Derartiges könnte man mir ihrer Datenbank doch schon seit Jahren machen, ist die Flexibilität der *Multitenant Database* sicherlich einmalig im Bereich der Datenbank Software. Mit ihrer Hilfe ist es jetzt möglich, dass die Datenbank schnell und vor allen Dingen flexibel auf unterschiedliche Anforderungen reagieren kann. Und das ist eine wesentliche Forderung, wenn es darum geht, einen Datenbank Service unter Cloud Gesichtspunkten aufzubauen.

Multitenant Database

Nach anderthalb Jahren hat sich die Multitenant Datenbank leider noch nicht wirklich am Markt durchgesetzt. Ein Grund dafür sind sicherlich die hohen Lizenzkosten aber ein weiterer ist die Unwissenheit um die Flexibilität dieser neuen Funktion. Zunächst einmal sieht alles viel komplizierter aus als vorher, ich muss mir über globale und lokale User oder Rollen Gedanken machen, es gibt Container, CDBs und PDBs und in der Version 12.1.0.1 war eine Pluggable Database noch nicht einmal geöffnet, wenn ich die Datenbankinstanz neu gestartet hatte, d.h. ich musste auch noch mit Startup Triggern arbeiten. Dann gab es auch noch die Aussage „Plug in gibt es nicht“ des Autors, was zwar im Prinzip richtig ist, aber nicht eben zur Vereinfachung beigetragen hat.

Der pragmatische Ansatz ist, wie so oft, beim Einsatz von Multitenant Database sicherlich der Beste: 12c (bitte 12.1.0.2) installieren, DBCA aufrufen und Checkbox „Create As Container Database“ anklicken. Nach kürzester Zeit hat man eine CDB als Rumpf der ganzen Architektur und eine oder besser zwei Pluggable Databases (PDBs). Zwei, weil es immer einen Template mit dem Namen PDB$SEED gibt, aus dem man neue PDBs erstellen kann, so man keine Migration aus einer Oracle 11g Datenbank vornehmen will. Natürlich kann man auch den Enterprise Manager oder auch Toad for Oracle (Database Browser) nehmen, und damit eine PDB anlegen, doch das Prinzip ist immer das gleiche: die Datendateien der Seed-Datenbank werden kopiert, umbenannt und schon habe ich eine neue PDB. Im Gegensatz zu älteren Releases dauert das Anlegen einer PDB nur noch wenige Minuten und wer auch noch über ein entsprechendes Storage (z.B. ACFS, ZFS, Direct NFS) verfügt, kann die Erstellung einer PDB durch Snapshot Copy auf Sekunden verkürzen.

Database As A Service

Wenn es um die Cloud Services geht, dann denkt man sicherlich zunächst an Amazon, Dropbox und ähnliche Anbieter, die es mir ermöglichen, Anwendungen (z.B. Microsoft Office 365), Plattformen (z.B. Oracle Cloud Services) oder nur Storage (z.B. Dropbox) im Internet zu nutzen. Auch wenn es immer noch viele Unternehmen gibt, die dem Prinzip skeptisch gegenüber stehen, so muss man ganz klar sagen, dass die Cloud Services (IaaS, PaaS, SaaS) nicht mehr wegzudenken sind. Im Vordergrund steht dabei nicht unbedingt die Nutzung über das Internet sondern die Preisgestaltung, da es nicht

mehr um eine „perpetual“ also einmalige immerwährende Nutzungsgebühr geht, sondern man zahlt, was man braucht. Nun ist es bei den meisten mir bekannten Unternehmen allerdings so, dass die Oracle Gebühren immer noch hoch und wenig flexibel sind. Einmalige Zahlungen sind eher die Regel als die Ausnahme. Doch die Unternehmen werden als Dienstleister für die internen oder externen Abteilungen immer mehr dazu gedrängt, ihre Services als monatliche Kosten mit einer größtmöglichen Flexibilität anzubieten. Für die Datenbank Infrastruktur bedeutet dies, dass der Service „Datenbank“ inklusive physischem Storage, RAM, CPU aber auch Konfigurationen wie Real Application Clusters, Replikation oder Data Guard flexibel angeboten und jeder Zeit anpassbar sein muss.

Database As A Service mit Multitenant Database

Die Multitenant Database kann an dieser Stelle der entscheidende Wettbewerbsvorteil sein. Natürlich verursacht er zunächst Kosten, doch diese können sich durch die Flexibilität und Schnelligkeit schnell amortisieren. Der Grund ist, dass eine Pluggable Database unabhängig von der verwendeten Konfiguration, also den HA oder DR Fähigkeiten ist. Wenn man also eine Infrastruktur aufbaut, die aus insgesamt 3 CDBs besteht, kann man die eigentlichen Datenbanken (also PDBs) frei auf diesen CDBs verteilen.

Beispiel:

Folgendes Beispiel soll dies veranschaulichen: ein größeres Unternehmen hat an jedem ihrer Standorte mehrere Bladecenter mit jeweils 8 Servern. Darauf verteilen sich diverse unterschiedliche Datenbanken. Die Hochverfügbarkeit wird dadurch gewährleistet, dass jeweils 1 Server nicht genutzt wird, so dass im Fehlerfall eine Datenbank bzw. die Instanz auf diesem „Spare“ Server betrieben werden kann. Fällt mehr als ein Server aus, wird auf ein zweites Bladecenter umgeschaltet, das über EMC SRDF angebunden ist. Insgesamt liegt damit die Auslastung eines Centers bei knapp über 40%. Wenn man außerdem den Storage-Anteil berücksichtigt (Raid-5 + SRDF), so liegt auch hier die Ausnutzung der Ressourcen bei gerade einmal 40%. Allerdings kommen alle Datenbanken, ob sie wollen oder nicht, in den Genuss einer Hochverfügbarkeit und Desaster Recoveries und muss von den Anwendungen auch so bezahlt werden.

Außerdem darf man an dieser Stelle auch die Entwicklungsabteilungen nicht vergessen. Für ein agiles Projektmanagement ist es erforderlich, dass mehrere Versionen einer Datenbank jederzeit zur Verfügung stehen. Bis zu 20 Datenbanken existieren daher für eine einzige Produktionsversion. Und um die Entwickler im Falle eines Serverausfalls nicht nach Hause schicken zu müssen, sind auch diese Datenbanken hochverfügbar ausgelegt.

Wie kann Multitenant Database helfen?

Zunächst einmal können die Bladecenter flexibler gestaltet werden. Dabei wäre es möglich, auf der Infrastruktur drei unterschiedliche Konfigurationen aufzubauen:

1. 4 Knoten für Hochverfügbarkeit (RAC) und Desaster Recovery (Data Guard) auf dem zweiten Bladecenter

2. auf jedem Bladecenter ein 2 Knoten RAC für Hochverfügbarkeit

3. auf jedem Bladecenter 4 Single Instance Datenbanken für unkritische Anwendungen.

Während man bei den unkritischen Anwendungen (4 Knoten) eine so genannte Single Tenant Database, d.h. eine CDB mit nur einer PDB (+SEED) aufbaut, und damit die Lizenzen für RAC sowie Multitenant spart, kann man auf den anderen Datenbanken jeweils eine CDB aufbauen. Dieser

einmalige Vorgang muss natürlich gut geplant und entsprechend getestet werden. Wenn die CDBs aber einmal aufgebaut sind, dann hat man für die Anwendung die größtmögliche Flexibilität.

Wenn eine Anwendung, die zunächst als unkritisch angesehen wurde, eine höhere Verfügbarkeit benötigt, dann kann sie einfach aus der Single Instance PDB „unplugged“ werden und in eine bereits vorhandene CDB entweder mit RAC oder über die zusätzliche Nutzung von Data Guard eingehängt werden. Es müssen dafür keine Änderungen an der Datenbank oder der Anwendung vorgenommen werden, außer dass der Service jetzt auf eine andere SCAN-Adresse zeigt.

Sollte sich im Laufe der Zeit zeigen, dass mehr HA und weniger „unkritische“ Datenbanken genutzt werden, kann der Cluster um einen Knoten erweitert werden und für die unkritischen Datenbanken stehen dann nur noch drei oder zwei Knoten zur Verfügung.

Und was ist mit dem Storage?

Schon seit einigen Jahren gibt es die Möglichkeit, den Storage durch Snapshot Copies oder Copy on Write (COW) effektiver zu nutzen. Dabei wird der Anwendung oder in unserem Fall der Datenbank vorgespielt, dass sie ihre eigenen Datafiles hat. Solange ein Datenblock nur gelesen wird, kann er von mehreren Datenbanken gemeinsam genutzt werden und erst beim Schreiben wird eine eigene Kopie des Blockes erstellt (daher Copy on Write). Gerade bei Entwicklungssystemen, die auf eine Kopie der Produktionsdaten (hoffentlich anonymisiert) angewiesen sind, können dadurch schnell Terabyte an Storage gespart warten. Prinzipiell ist dies schon seit Version 11.2.0.3 mit „CloneDB“ möglich, allerdings wird dafür zunächst eine 1:1 Kopie in Form eines RMAN Backup Copies benötigt. Mit Oracle 12c kann man eine PDB als Source für ein Snapshot Copy verwenden und dann beliebig viele (bis zu 251) PDBs erstellen, die bei lesenden Zugriffen auf diese Kopie zugreifen. Voraussetzung ist dafür zunächst ein Storage, was Snapshot Copies erlaubt (d.h. allen voran Oracle ZFS) aber man kann auch ACFS nehmen, d.h. die Datenbank statt im ASM im ASM Cluster Filesystem betreiben. Ab 12.1.0.2 ist es jetzt auch möglich, mit einem NFS Server und DirectNFS einen PDB Snapshot Clone aufzubauen. Somit ist hier eine kostengünstige und hardwareunabhängige Lösung möglich.

Fazit

Mit der Multitenant Database und dem Snapshot Copy für die Pluggable Databases ist es jetzt endlich möglich, sehr flexibel auf die wechselnden Anforderungen von Anwendungen einzugehen. Ob es sich dabei um Entwicklungssysteme oder hochkritische Datenbanken geht, ist dabei unerheblich und gerade der Wechsel von einer Anwendungsklasse (z.B. „Unkritisch“) in eine andere (z.B. „Business Important“ oder „Business Critical“) kann innerhalb weniger Stunden und ohne Änderung der Datenbank durchgeführt werden. Damit kann man das Fragezeichen in der Überschrift getrost durch ein Ausrufezeichen ersetzen. Multitenant Database ist DIE Lösung für DBaaS.

Kontaktadresse:

Johannes Ahrends
CarajanDB GmbH
Siemensstraße 25
D-50374 Erftstadt
Telefon: +49 (22 35) – 1 70 91 84
Fax: +49 (22 35) – 1 70 89 79
Mobil: +49 (1 70) - 4 05 69 36
E-Mail johannes.ahrends@carajandb.com
Internet: www.carajandb.com