

# Warum sollte man die Multitenant Database Option verwenden?

**Johannes Ahrends  
CarajanDB GmbH  
Erfstadt**

## **Schlüsselworte**

Multitenant Database. Oracle 12.2, Schema Konsolidierung, Private Cloud, Virtualisierung

## **Einleitung**

Seit 3 Jahren gibt es die Multitenant Database Option jetzt und die Implementierungsrate ist immer noch sehr gering. Das mag zum einen an der generellen Haltung gegenüber dem neuen Release 12c handeln aber ein sicherlich entscheidender Faktor ist, wie sollte es auch anders sein, die Kosten. Mit \$ 17.500,00 pro Prozessor (Oracle Technology Global Price List August 18, 2016) gehört die Option zu den teuersten Erweiterungen der Datenbank. Sollte man das das Geld nicht lieber in die Advanced Security Option stecken, die ist sogar noch etwas preiswerter. In dem Vortrag lege ich Ihnen die Gründe dar, die bei einem meiner derzeitigen Projekte dazu geführt haben, die Multitenant Database Option einzusetzen.

## **Ausgangssituation**

In dem Projekt sollte die bestehende Hardwareinfrastruktur basierend auf einem 6 Knoten Stretched RAC Cluster (zwei Standorte) modernisiert werden. Neben dem Austausch der Hardware standen allerdings noch ein Punkt im Raum: Die Anwendungen sollten flexibler werden, d.h. die bisher betriebene Schemakonsolidierung sollte aufgebrochen werden. Eine erste Überschlagsrechnung ergab, dass aus den derzeitig ca. 10 Datenbanken insgesamt 70 – 80 entstehen würden.

Es war keine Frage, dass dafür nur eine Lösung in Frage kommt: Virtualisierung! Größtmögliche Flexibilität bei geringen Kosten und hoher Verfügbarkeit – was will man mehr?

Leider gibt es, wie sicherlich alle DOAG Mitglieder wissen, ein kleines Problem beim Betrieb von Oracle Datenbanken auf virtuellen Umgebungen: die Lizenzierung. Natürlich ist das kein echtes Problem, wenn man OracleVM einsetzt. In der Praxis wird allerdings, und so auch in diesem Projekt, VMware eingesetzt und selbst mit der auf der DOAG Konferenz und Ausstellung 2015 getroffenen Vereinbarung mit Andy Mendelsohn sind die Bauchschmerzen bei den Lizenzverantwortlichen in den Unternehmen was die Bedingungen angeht, groß.

In dieser Phase des Projektes wurde ich gebeten, einmal die Multitenant Database Option vorzustellen. Zwar konnte ich die Zweifel nicht ganz ausräumen, denn es fehlt schlicht und einfach die Erfahrung in Bezug auf den Einsatz der Option, aber insgesamt überwogen die Vorteile dieser Architektur. Damit kann jede Anwendung seine eigene Datenbank bekommen, ohne jedoch viele kleine Datenbanken auf einem bzw. mehreren Servern administrieren zu müssen. Die Anwendungen können einzeln migriert werden, trotzdem muss nicht für jede Anwendung eine eigene Datenbank Software installiert werden.

## Projektstand

Als Architektur haben wir uns letztendlich für einen zwei-Knoten Real Application Clusters entschieden, die über Dataguard (ebenfalls RAC) abgesichert wird. Wir werden zunächst ca. 7 CDBs auf dieser Architektur betreiben. Gründe für die Anzahl von CDBs sind:

1. Jede Organisationseinheit (insgesamt 3) möchte seine „eigene“ Datenbank
2. Es gibt die Anforderung für zwei Zeichensätze (WE8ISO8859P15, AL32UTF8)
3. Die größte Datenbank möchte man komplett isolieren (CDB mit einer PDB)

Sicherlich ist das nicht optimal, es werden immer noch viele Ressourcen durch die CDBs unnötig verschwendet, aber es ist ein erster Schritt.

Die komplette Architektur wird viermal identisch (bis auf CPU und RAM) aufgebaut:

1. Maintenance: „Spielwiese“ für die Administratoren
2. Test: Eigentlich die Entwicklungsumgebung
3. Vorproduktion: Natürlich braucht man eine Umgebung, in der Patches, neue Releases, etc. erst einmal getestet werden können.
4. Produktion

Eine wichtige Forderung beim Aufbau der Architektur war ja, dass die Anwendungen eine höhere Flexibilität erhalten sollen. Damit ist unter anderem gemeint, dass die Möglichkeit gegeben sein muss, eine einzelne Anwendung zurückzurollen. Leider ist das im Release 12.1 für eine einzelne PDB nicht möglich. Also was tun?

Wir haben hier sogar zwei Lösungen:

1. Wir können jederzeit die Standby Datenbank mit der Funktion „Snapshot Standby“ in eine „normale“ Datenbank überführen und mit Flashback Database auf einen älteren Stand zurücksetzen. Dann können wir eine PDB mit Unplug / Create aus dieser Datenbank herausholen und in die Produktion einspielen. Der Zeitaufwand ist dabei proportional zur zu übertragenen Datenmenge.
2. Als Storage setzen wir NetApp mit den Snapshot Mechanismen ein. Damit können wir in wenigen Minuten die gesamte CDB klonen und uns daraus bedienen.

Trotzdem wäre es natürlich wünschenswert, die Flashback Database Funktion auch in der PDB zu haben (wait and see).

## Offene Punkte

Wir möchten, weil es in dem Unternehmen seit Jahren üblich ist, Oracle Managed Files verwenden. Leider gibt es hier noch einige „Herausforderungen“. So haben wir z.B. für bestimmte PDBs eigene Filesysteme vorgesehen. Leider wird das beim Duplizieren der CDB mit RMAN nicht erkannt, d.h. alle PDBs landen im gleichen (falschen) Verzeichnis. Ähnliches gilt auch für die Erstellung der PDBs nach dem Kopieren. Hier wird zwar der Parameter „CREATE\_FILE\_DEST“ erkannt, allerdings der DB\_UNIQUE\_NAME nicht! Das führt dazu, dass die PDB auf der Standby Seite nicht erstellt werden kann, weil logischerweise hier der DB\_UNIQUE\_NAME unterschiedlich zur primären Datenbank ist.

## **Fazit**

Sicherlich gibt es noch einige Einschränkungen bei der Verwendung der Multitenant Database Option und ich würde mir wünschen, dass Oracle die Kunden an der Stelle nicht im Regen stehen ließe. Aber es zeigt sich, dass es, so man nicht die Gratwanderung mit der Virtualisierung gehen möchte, keinen Weg an der Multitenant Datenbank vorbei gibt. Wenn man sich allein überlegt, wie viel Zeit ein DBA dafür verwenden muss, Exports zu erstellen, weil wieder eine Anwendung die „Flexibilität“ haben muss, zurückgerollt zu werden. Oder das Thema Patching. Wie will man 70 Datenbanken mit je einem eigenen Oracle Home innerhalb eines Quartals (inklusive Testen!) überhaupt patchen? Sicherlich werden bis November noch weitere interessante Erfahrungen dazu kommen.

### **Kontaktadresse:**

Johannes Ahrends  
CarajanDB GmbH  
Siemensstraße 25  
D-50374 Erftstadt

Telefon: +49 (0) 2235 – 170 91 84  
Fax: +49 (0) 2235 – 170 79 78  
E-Mail: [johannes.ahrends@carajandb.com](mailto:johannes.ahrends@carajandb.com)  
Internet: [www.carajandb.com](http://www.carajandb.com)