

# Service, please!

**Jochen Kutscheruk**  
**merlin.zwo InfoDesign GmbH & Co. KG**  
**76228 Karlsruhe**

## Schlüsselworte

Bitte schreiben Sie hier Ihre Schlüsselwörter.

## Einleitung

Services sind ein viel zu selten genutztes Feature der Oracle-Datenbank. Klug eingesetzt können sie dem Administrator die Arbeit deutlich erleichtern.

Ich möchte darstellen, wie Datenbank, Instanz und Service zusammenhängen und wie man über die Strukturierung in Services die Kontrolle über die Datenbank behalten kann. Von der Single Instance über RAC und DataGuard bis hin zum Ressource Manager können und sollten Services immer eingesetzt werden. Anhand von praktische Beispielen werden die Vorteile und Einsatzmöglichkeiten von Services erläutert.

## Was ist ein Oracle Datenbank Service?

Jede Oracle Datenbank besteht zuerst aus physischen Dateien:

- Datafiles
- Online-Redologs
- Control Files
- Archive Logfiles
- Parameter File
- Passwort File

Die Daten liegen in den Datenbankfiles, sind jedoch von außen so nicht zugänglich. Dazu wird eine Datenbank-Instanz, also ein Programm, benötigt.

Diese Datenbank-Instanz verwaltet die Datenbank und legt dazu im Hauptspeicher des Servers entsprechende Verwaltungsstrukturen an, z.B. Buffer Cache, Shared Pool, Log Buffer, Large Pool, Streams Pool, Java Pool, um nur die wichtigsten zu nennen.

Zusätzlich werden von dieser Instanz auf dem Server weitere Prozesse gestartet, z.B. SMON, PMON, DBWn, CKPT, LGWR, ARCn.

Über diese Instanz, das eigentliche RDBMS, kann jetzt per SQL auf die in der Datenbank enthaltenen Daten zugegriffen sowie DML und DDL – Operationen ausgeführt werden.

Dies geht bis zu diesem Zeitpunkt jedoch nur lokal auf dem Server per Zugriff über den Shared Memory.

Damit man auch von Extern auf die Datenbank zugreifen kann benötigt man einen Listener. Dieser lauscht im Netzwerk nach Verbindungsanfragen zu der Datenbank, nimmt diese entgegen und startet auf dem Server einen Prozess, über den der entfernte Rechner dann direkt mit der Datenbank kommunizieren kann.

Und genau hier beim Listener taucht zum ersten Mal der Begriff des „Services“ auf:

```
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=m2doag)(PORT=1521)))
Services Summary...
Service "doag" has 1 instance(s).
  Instance "doag", status READY, has 1 handler(s) for this service...
Service "doagXDB" has 1 instance(s).
  Instance "doag", status READY, has 1 handler(s) for this service...
```

Abb 1: Der Service im Listener

Das externe Programm, welches eine Verbindung zu der Datenbank herstellen will, muss beim Verbindungsaufbau angeben, auf welchem Server (IP-Adresse oder Name) es sich mit welcher Datenbank verbinden möchte.

Der Listener auf dem Server überprüft dann, ob ihm diese Datenbank bekannt ist und stellt die Verbindung her.

Die Identifizierung der Datenbank geschieht dabei entweder über die SID, also den Instanznamen, oder aber über den Servicennamen der Datenbank. Dieser Servicename entspricht dabei der SID, eventuell ergänzt durch den Domainnamen der Datenbank (Parameter DB\_DOMAIN).

Ein Servicename dient also – vereinfacht gesagt – der eindeutigen Identifizierung einer Instanz und damit einer Datenbank.

### Wo werden Services verwendet?

Services werden standardmäßig bei Oracle RAC verwendet. Der Grund ist einfach: bei Oracle RAC wird eine Oracle Datenbank von mehreren Instanzen geöffnet. Jede Instanz hat dabei einen eigenen Instanznamen, also SID.

Nehmen wir eine RAC-Datenbank mit Namen „DOAG“. Diese Datenbank wird von beispielsweise 3 Instanzen – auf Server1, Server2 und Server3 – geöffnet. Per Default heißt dann die Instanz auf Server1 „doag1“, die auf Server2 „doag2“ und die auf Server3 „doag3“. Dies gilt für eine Admin-managed Installation, bei Policy-managed heißen die Instanzen dann „doag\_1“, „doag\_2“ und „doag\_3“. Bei der Verbindung auf die Datenbank müsste man folglich immer die Server-IP und die SID der Instanz angeben, um sich auf die Datenbank zu verbinden.

Da dies offensichtlich nicht mit dem Konzept der Hochverfügbarkeit vereinbar ist, läuft auf allen drei Instanzen der Service „doag“. Dieser Service ist sowohl dem lokalen Listener als auch dem übergeordneten („knotenübergreifenden“) SCAN-Listener bekannt. Der Client verbindet sich daher zum SCAN-Listener unter Angabe des gewünschten Service und wird von diesem dann eine passende Instanz verwiesen, welche diesen Service zur Verfügung stellt.

Bei einer RAC-Datenbank gibt es immer einen Service, welcher dem Datenbanknamen entspricht. Dieser Service läuft auch automatisch auf allen Knoten der RAC-Datenbank. Man kann jedoch jederzeit weitere Services für diese Datenbank dazu definieren („srvctl add service ...“), welche unterschiedliche Eigenschaften haben können.

Z.B. muss ein Service nicht auf allen Knoten gleichzeitig laufen. Man könnte für die Datenbank „doag“ einen Service „doag\_oltp“ definieren, welcher nur auf Knoten 1 und 2 läuft, während ein Service „doag\_dwh“ lediglich auf Knoten 3 läuft. Dadurch könnte ganz einfach die Auswertungslast

von DWH-Abfragen auf Knoten 3 gelegt werden, wohingegen Knoten 1 und 2 vollständig für den OLTP-Betrieb zur Verfügung stehen.

Für verschiedene Services können unterschiedliche Parameter gesetzt werden. Ein Service kann zum Beispiel speziell für Webanfragen konfiguriert werden (Connection Load Balancing Goal: SHORT), während ein anderer Service für klassische Client-Server Verbindungen konfiguriert wird (Connection Load Balancing Goal: LONG). Dies steuert das Verhalten, wie die eingehenden Verbindungen auf die einzelnen Knoten verteilt werden.

Ebenso werden Services bei Dataguard-Konfigurationen verwendet. In einer Dataguard-Konfiguration existiert eine Datenbank n-mal auf verschiedenen Servern. Eine dieser Datenbanken ist die aktuelle Primärdatenbank (diese ist aktiv), während alle anderen Kopien den Status „Physical/Logical Standby“ haben. Diese Standby-Datenbanken sind entweder im Mount-Status oder aber „Read Only“ bzw. „Read Only With Apply“ (bei Active Dataguard).

Alle diese Datenbanken haben jedoch den gleichen Namen und gegebenenfalls auch die gleiche SID, so dass sie bereits ab dem Mount-Status als Service im Listener erscheinen. Der Client kann daher nicht direkt erkennen, auf welchem Server sich gerade die aktive Primärdatenbank befindet.

Daher wird für den Client, z.B. in der tnsnames.ora, eine Liste der möglichen Datenbankserver hinterlegt, zusammen mit einem Servicennamen, welcher nur auf der Primärdatenbank läuft. Dadurch kann sich der Client bei einem Dataguard Switchover oder Failover automatisch wieder auf die gerade aktuelle Primärdatenbank verbinden, ohne dass irgendwelche Änderungen an der Konfiguration notwendig wären.

Sofern auf den einzelnen Servern jeweils die Oracle Clusterservices installiert sind, wird der Service über diese Clusterservices aktiviert, sobald die Datenbank als Primärdatenbank gestartet wird. Ansonsten muss der Service über einen Startup-Trigger in der Datenbank aktiviert werden, der den Service startet, sobald die Datenbank im Read/Write Modus geöffnet wird.

### **Warum sollte ich Services verwenden?**

Bei den beiden Anwendungsszenarien RAC und Dataguard müssen Services verwendet werden, damit die Konfiguration korrekt funktioniert.

Daneben sollten jedoch Services selbst bei einer Single-Instance Datenbank als Standard genutzt werden. Grundsätzlich sollte dabei für jeden Anwendungsbereich ein eigener Service definiert werden, also z.B. ein Service für Fibu, für Lohn und für Auswertungen. Es ist sogar sehr hilfreich, selbst für nächtliche Batchläufe eigene Services zu definieren.

Über die passenden Services für die Datenbank können Sie alle Vorteile nutzen, die sich daraus ergeben:

- Bei RAC die gewünscht Failover-Policy nutzen
- Bei RAC das passende Loadbalancing einstellen
- Bei Dataguard automatisch den Client der aktiven Primär- oder Active Dataguard – Datenbank folgen lassen
- Mit Hilfe des Service können Sie erkennen, welche Programme welche Last auf der Datenbank verursachen

- Sie können Services gezielt disablen – inklusive Verbindungstrennung der noch aktiven Sessions auf diesem Service – um z.B. Updates für ein Programm störungsfrei einspielen zu können, während die Datenbank insgesamt ungestört weiterläuft
- Sie können ein Amok laufendes Programm problemlos aus der Datenbank aussperren, ohne dass andere Programme davon beeinträchtigt werden
- Sie können Services selbst bei manuellen Standby-Datenbanken nutzen, damit sich der Client automatisch auf die richtige aktive Datenbank verbindet – ohne etwas an der Konfiguration ändern zu müssen
- Besonders einfach ist die Nutzung des Resource Managers (Enterprise Edition) bei der Verwendung von Services. Hierüber können Sie den Services unterschiedliche Prioritäten und Ressourcen zuweisen – sei es CPU, I/O, Anzahl der Verbindungen, Zwangstrennung, ... Dies ist extrem nützlich, um z.B. Reports oder Auswertungen, die zu viel Ressourcen beanspruchen und damit den OLTP-Betrieb stören, niedriger zu priorisieren.
- Sie können einen Service mit einer Edition koppeln, so dass Sie die Connections ganz einfach auf die richtige Edition leiten können.
- ....

### **Zusammenfassung**

Bereits seit Version 8 empfiehlt Oracle die Nutzung von Services statt der SID bei der Verbindung. Leider sieht man auch heute noch – 20 Jahre später – Programme, die sich nur auf die SID verbinden können, da man in der Konfiguration nichts Anderes einstellen kann. Dies macht die Nutzung der Features von RAC und Dataguard quasi unmöglich. Ebenso ist die Nutzung neuer Features, wie z.B. des GSD, nicht möglich, da dieses ausschließlich über Services arbeitet.

Services können sehr einfach in beliebiger Zahl je Datenbank eingerichtet werden. Es ergeben sich daraus nur Vorteile, Nachteile sind mir bisher keine bekannt geworden.

Nutzen Sie Services in der Datenbank, es erleichtert das Leben als Administrator ungemein!

### **Kontaktadresse:**

Jochen Kutscheruk  
 Merlin.zwo InfoDesign GmbH & Co. KG  
 Elsa-Brändström-Straße 14  
 D-76228 Karlsruhe

Telefon: +49 (0) 721-132 096 0  
 Fax: +49 (0) 721-132 096 99  
 E-Mail: jochen.kutscheruk@merlin-zwo.de  
 Internet: www.merlin-zwo.de