

Effizienter Betrieb von Oracle Single-Instanzen

Rastislav Ciganek
Festo AG & Co. KG, Esslingen

Michael Günzel
Festo AG & Co. KG, Esslingen

Schlüsselworte

Oracle Datenbank, Betrieb, Services, Tablespace Point-in-Time-Recovery (TSPITR), interne Verrechnung

Einleitung

Viele „kleinere“ Applikationen bzw. deren Schemas werden oft in dedizierten Oracle Instanzen gehostet, neuerdings auch in kostenpflichtigen s.g. pluggable Databases. Das kann zu einer größeren Anzahl von Oracle Datenbanken führen. Dadurch entsteht ein großer Overhead im Bereich der DB-Administration.

Auf konsolidierten Single-Instanzen lassen sich unter Beachtung bestimmter Bedingungen auch mehrere Applikationen sehr elegant betreiben.

Durch eine geeignete Schemagruppierung -anhand der Kritikalität und den wichtigsten Datenbankeigenschaften- werden einige bekannte, in der Praxis aber wenig verwendete Techniken vorgestellt. Zum Einsatz kommt die Tablespace Point-in-Time-Recovery Analyse und die Einführung von Service Namen mit Host-Aliasen, sowohl bei Single-Instanzen als auch bei Dataguard-Umgebungen.

Durch den Einsatz der vorgestellten Techniken, können Oracle Datenbank Betreiber eine möglichst überschaubare Anzahl von Oracle Instanzen für viele Anwendungen deutlich (kosten)effektiver nutzen.

Kurzes Unternehmensporträt der Firma Festo

Im Zuge der Digitalisierung richtet sich Festo als weltweit führender Anbieter von Automatisierungstechnik und technischer Bildung mit seinen Produkten und Services auf die smarte Produktion der Zukunft aus. Seit über 60 Jahren ist das 1925 gegründete unabhängige Familienunternehmen mit Sitz in Esslingen a.N. Impulsgeber in der Automatisierung und hat sich mit einem einzigartigen Angebot zum Weltmarktführer in der technischen Aus- und Weiterbildung entwickelt. 300.000 Kunden in der Fabrik- und Prozessautomation weltweit vertrauen auf die pneumatischen und elektrischen Antriebslösungen des Unternehmens. Festo Didactic bietet darüber hinaus modernste Qualifizierungslösungen für Industrieunternehmen und Bildungseinrichtungen in aller Welt. Die Festo Gruppe erzielte im Geschäftsjahr 2016 einen Umsatz von 2,74 Mrd. Euro und ist mit 18.800 Mitarbeiterinnen und Mitarbeitern an 250 Standorten weltweit präsent. 8 % vom Umsatz fließen jährlich in die Forschung und Entwicklung.

Als Familienunternehmen denkt und handelt Festo langfristig und verantwortungsbewusst. Festo steht für klare Werte, höchste Qualität und kundenorientierte Innovation. In der industriellen Automatisierungstechnik und technischen Bildung setzt Festo seit seinen Anfängen Maßstäbe und leistet damit einen Beitrag zu einer nachhaltigen Entwicklung in Wirtschaft und Gesellschaft.

Die Digitalisierung treibt Festo in allen Unternehmensbereichen voran. Eine Vielzahl der Aspekte von Industrie 4.0 ist im Unternehmen Festo heute bereits Realität. Festo begleitet seine Kunden und Mitarbeiter in die digitale Zukunft. Dafür entwickelt das Unternehmen neue zukunftsweisende Konzepte, die auf den Dreiklang von innovativen und energieeffizienten Technologien, intuitiver Mensch-Maschine-Kollaboration sowie Aus- und Weiterbildung setzen.

Die massiv steigende Quantität an Daten und Komplexität in der Vernetzung von Anlagen- und Unternehmensbereichen macht auch vollkommen neue und hochflexible Lösungen zur Qualifikation von Mitarbeitern in der Industrie und Auszubildenden in Berufsschulen nötig. Für den Betrieb und die ständige Optimierung von Anlagen, die Erreichung einer hohen Wandelbarkeit und die Entwicklung von flexiblen intelligenten Komponenten müssen ein generelles Verständnis für digitale, vollautomatisierte Produktionstechnologien und eine tiefgehende Kenntnis im Aufbau und der Programmierung von digitalen Anlagen-Netzwerken vermittelt werden.

DB-Team Organisation

Aktuell betreibt das internationale virtuelle DBA Team u.a. das Datenbankprodukt Oracle Datenbank Enterprise Edition als Single Instanzen oder bei mission critical Systemen mit Dataguard. Auf konsolidierten Oracle Datenbanken betreiben wir eine größere Anzahl Applikationen. Die Applikationsschicht wird von unterschiedlichsten Abteilungen betreut. Deswegen versucht man vereinfachte Methoden zu finden, welche es dem DBA Team erlaubt wenig invasiv bei Oracle DB-Verbindungseinstellungen zu agieren.

Folgende Möglichkeiten nehmen wir als Voraussetzung für unser Konzept:

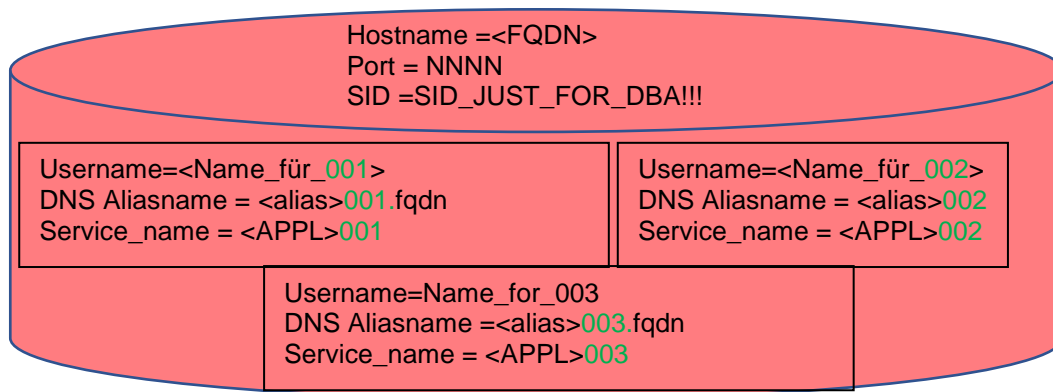
1. DBA Team soll selbständig oder zumindest sehr schnell und unkompliziert Host-Aliase für die Datenbank Hosts zentral in einer Domain definieren bzw. beantragen können.
2. Die Anpassung der spfile- oder Startup-Trigger- Erstellung muss in der Oracle DB Umgebung möglich sein.
3. Optional ist zu empfehlen eine zentrale TNS-Alias-Vergabe zu pflegen. Dazu kann man bspw. eine zentrale tnsnames.ora Datei(en) pflegen oder Oracle Internet Directory bzw. ähnliches LDAP Produkt zwecks TNS-Aliaspflege betreiben.

Einführung der Oracle Service Namen und Host-Aliassen

Die Hauptmotivation für die Einführung von Oracle Service Names in den Oracle Datenbanken und auch Einführung der Host-Aliassen war eine möglichst einfache und „lebenslange“ Einhaltung der DB Zugangsdaten (Host, Port, Servicename) für die einzelnen Applikationen. Dazu benötigen wir im einfachsten Fall die Anpassung des Initialisierung-Parameters `service_names` mit neuen Servicennamen bspw. `APPL001` und `APPL002`, wie im Beispiel unten skizziert. Diese neuen Servicennamen sollten mit neuerstellten Host-Aliassen bspw. `alias001` und `alias002` in der Dokumentation oder im DNS Tool im Kommentar korrespondieren.

```
alter system set service_names= <DEFAULT>, APPL001, APPL002 scope=BOTH;
```

Auf Basis dieser Definition kann man sich eine Reihe von unterschiedlichen Zugängen für eine Single-DB anlegen und je nach Bedarf den Client Connectstring entsprechend definieren. Anbei die gängigsten von vielen Varianten.



```
TNS_alias_for_001 =
  (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)
                        (HOST = alias001.fqdn)(PORT=<NNNN>))
    (CONNECT_DATA =(SERVICE_NAME = APPL001)))
```

```
OCI JDBC Driver
jdbc:oracle:oci:@alias001.fqdn:<NNNN>/APPL001
```

```
Thin JDBC Driver
jdbc:oracle:thin:Name_for_001/Pwd_for_001@//alias001.fqdn:<NNNN>/APPL001
```

Für eine Oracle Dataguard Umgebung benötigen wir je DB-Host extra Aliase bspw. alias001 für die Primary und alias501 für die Secondary. Die Oracle Service Names Definition bspw. APPL001_501 muss durch das Oracle Package dbms_service angelegt werden. Ein „Startup-Trigger“ ist notwendig. Anbei ein vereinfachtes Beispiel dazu.

```
exec DBMS_SERVICE.CREATE_SERVICE ('APPL001_501','APPL001_501');
exec DBMS_SERVICE.START_SERVICE ('APPL001_501');
```

```
CREATE OR REPLACE TRIGGER STARTUP_SERVICES
  after startup on database
DECLARE
...
  IF database_role = 'PRIMARY' THEN
      DBMS_SERVICE.START_SERVICE ('default_service');
      DBMS_SERVICE.START_SERVICE ('APPL001_501');
  ELSE
      DBMS_SERVICE.START_SERVICE ('otherservice_for_readonly');
  END IF;
END;
/
```

```
TNS_alias_for_001_501 =
  (DESCRIPTION=(ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=TCP)(HOST=alias001)(PORT=<NNNN>))
    (ADDRESS=(PROTOCOL=TCP)(HOST=alias501)(PORT=<NNNN>)))
  (CONNECT_DATA=(SERVICE_NAME=APPL001_501)))
```

Vorteile der Oracle Service Namen und Host-Aliassen

- DB-Zugang ist „unabhängig“ von der Oracle SID oder „allgemeingültigen“ Service_names
- DB-Hosting ist dadurch deutlich flexibler
- Migrationen werden viel einfacher
- Ressourcenzuweisung mittels Ressource Manager ist möglich
- Das Tracing einer Applikation mit mehreren Connections ist einfacher auszuwerten
- Kleinere Anzahl der Oracle Instanzen führt zu niedrigerem Overhead bei der Virtualisierungsschicht, dem Backup-Management und Monitoring-Aufwänden.

Analyse der Oracle Enterprise Datenbanken mit Tablespace Point-in-Time-Recovery (TSPITR)

Auf den shared Datenbanken werden unterschiedlichsten Unternehmensapplikationen in getrennten Schemas und Tablespaces gehostet. Es wäre sehr wünschenswert zumindest in Oracle Enterprise Edition ohne Options eine vereinfachte Möglichkeit für ein „Schema-Point-in-Time-Recovery“ zu realisieren. Wir haben die Methode Tablespace Point-in-Time-Recovery näher analysiert. Als Beispiel für eine schnelle und vereinfachte TSPITR Analyse kann man die System Views SYS.TS_PITR_CHECK und SYS.TS_PITR_OBJECTS_TO_BE_DROPPED verwenden. Diese zeigt uns, welche Schema-Objekte aktuell nicht wiederherstellbar sind und ab welchem Zeitpunkt.

Anbei noch ein paar Beispiele für Abfrage und Befunde im SYS.TS_PITR_CHECK.

```
SELECT *
FROM SYS.TS_PITR_CHECK
WHERE ( 'SYSTEM' IN (TS1_NAME, TS2_NAME)
      AND TS1_NAME <> TS2_NAME
      AND TS2_NAME <> '-1'
      AND OBJ1_OWNER not in
        ('SYS', 'SYSTEM', 'CTXSYS', 'GSMADMIN_INTERNAL', 'WMSYS', 'XDB')
      )
OR ( TS1_NAME not in ('SYSTEM', 'SYS', 'SYSAUX', 'TEMP')
    AND TS2_NAME = '-1'
    AND OBJ1_OWNER not in
      ('SYS', 'SYSTEM', 'CTXSYS', 'GSMADMIN_INTERNAL', 'WMSYS', 'XDB')
    );

select owner, to_char(max(creation_time), 'DD.MM.YYYY HH24:MI:SS')
TSPITR_possible_until
  from SYS.TS_PITR_OBJECTS_TO_BE_DROPPED
 where tablespace_name not in ('SYSTEM', 'SYS', 'SYSAUX', 'TEMP')
    and owner not in
      ('SYS', 'SYSTEM', 'CTXSYS', 'GSMADMIN_INTERNAL', 'WMSYS', 'XDB')
 group by owner
 order by 1
```

Beispiel für nicht erlaubte Objekte	Fehlermeldung im View
INDEX <Indexname> ON <Tablename> (<Function>("<column>"),	Functional Indexes not supported
(Oracle Text) TABLE DR\$<>\$R (ROW_NO, DATA BLOB)	Secondary Objects not allowed in Recovery Set
CREATE TABLE <tablename> ...; CREATE SNAPSHOT <snapname> as select * from <tablename>;	Master table used for snapshots not allowed in recovery set
(Oracle Text) INDEX . (INDEXTYPE IS CTXSYS.CONTEXT)	Tables and associated indexes not fully contained in the recovery set
CREATE SNAPSHOT MATERIALIZED VIEW	Snapshots not allowed in recovery set
CREATE USER <username> ... DEFAULT TABLESPACE <tbls_name> .. ; CREATE TABLE <Tablename> Missing default tablespace; Create some constraints;	constraint between tables not contained in recovery set

Unser Fazit nach der Analyse: In einer historisch gewachsenen Umgebung ist es erstmal nicht möglich eine einfaches „Schema-Point-in-Time-Recovery“ zu realisieren. Dafür wäre ein Redesign der Applikation nötig. Bsp. keine Function-Based Indexes, Snapshots verwenden oder Oracle Text feature deaktivieren.

Vor- und Nachteile von Shared-Datenbanken

Vorteile:

- Shared-Datenbanken verringern den administrativen Aufwand (Patching, Upgrades, ...). Dieser DB-Typ ist bei Festo -ohne Servicennamen- bereits seit 10 Jahren im Einsatz.
- Verrechneter Preis bzgl. des Fachbereichs ist deutlich niedriger, aufgrund geteilter Lizenz- und Wartungskosten im Vergleich zu einer Single Instance.
- Ermöglicht schnelles Deployment. Ein späterer Umzug auf eine dedizierte Umgebung ist einfach möglich. Keine Aufwände auf der Applikationsseite (Client-Einstellungen) durch die Verwendung des Service-Name Konzepts.

Nachteile:

- Garantierter Restore Point auf Instanz-Ebene ist kein geeignetes Mittel, um Wartungsarbeiten an einzelnen Schemas abzusichern.
- Tablespace Point In Time Recovery (TSPITR) ist nicht immer möglich.
- Noisy Neighbours beeinflussen andere Applikationen. Abhilfe kann der Ressource-Manager schaffen.

Flexibilisierung durch Service-Name-Konzept:

Die aktuelle Linux-Migration war durch die die vorhandenen Service-Namen ohne Konfigurationsaufwände auf Applikationsseite möglich.

Kontaktadresse:

Rastislav Ciganek, Festo AG & Co. KG
E-Mail rastislav.ciganek@festo.com
Michael Günzel, Festo AG & Co. KG
E-Mail michael.guenzel@festo.com