

Connecting and using RAC

Martin Schmitter



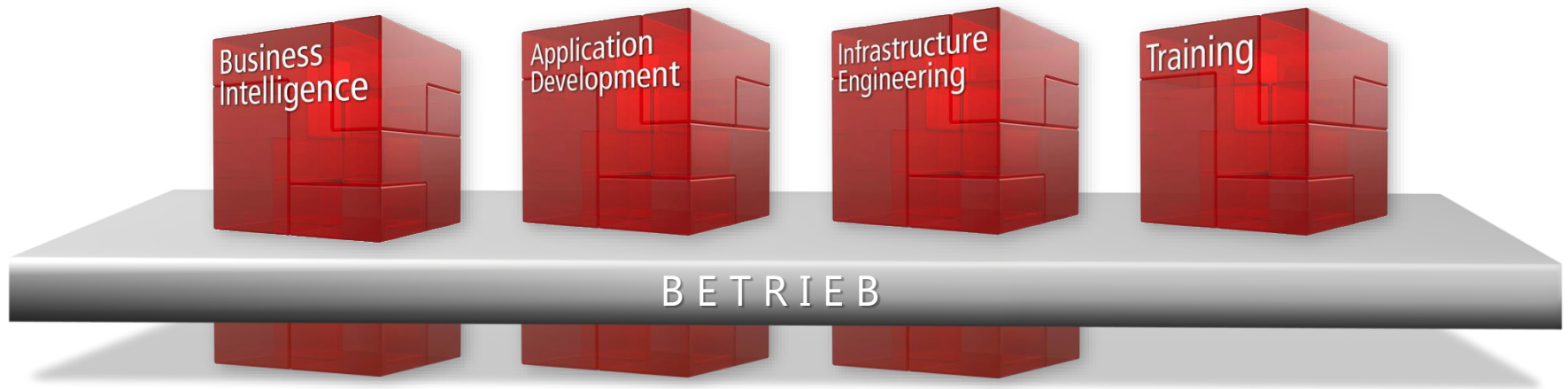
Trivadis
makes IT
easier.

BASEL BERN BRUGG LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

■ Unser Unternehmen

Trivadis ist **führend bei der IT-Beratung, der Systemintegration, dem Solution-Engineering** und der Erbringung von **IT-Services** mit Fokussierung auf **ORACLE®** und  **Microsoft** Technologien im D-A-CH-Raum.

Unsere Leistungen erbringen wir aus den strategischen Geschäftsfeldern:



Trivadis Services übernimmt den korrespondierenden Betrieb Ihrer IT Systeme.

■ Mit über 600 IT- und Fachexperten bei Ihnen vor Ort



12 Trivadis Niederlassungen mit über 600 Mitarbeitenden

200 Service Level Agreements

Mehr als 4'000 Trainingsteilnehmer

Forschungs- und Entwicklungsbudget: CHF 5.0 / EUR 4 Mio.

Finanziell unabhängig und nachhaltig profitabel

Erfahrung aus mehr als 1'900 Projekten pro Jahr bei über 800 Kunden

■ Über mich ...

- Senior Berater, Trivadis GmbH
 - Standort Düsseldorf
- Erfahrung mit Oracle Produkten seit 1998:
 - Administration & Support
 - Architektur
 - Firefighting
 - Projektunterstützung
- Schwerpunkte:
 - Hochverfügbarkeit (RAC)
 - Enterprise Manager (Monitoring)
 - Oracle, auch mit Windows
 - Cloud Computing



Server Side Load Balancing

Fast Application Notification - FAN
Realtime Connection Load Balancing

Fast Connection Pool

Oracle Data Provider for .Net
Load Balancing
Connection Time Load Balancing

Client Side Load Balancing

Transparent Application

Universal Connection Pool

Oracle Notification Service - ONS
Application Failover - TAF

Schon mal gehört?

Technik allein bringt Sie nicht weiter. Man muss wissen, wie man sie richtig nutzt.



BASEL BERN BRUGG LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

■ AGENDA

1. Übersicht
2. Load Balancing & Failover
3. DB Services
4. FAN/FCF
5. ODP und UCP
6. Zusammenfassung



Übersicht

■ Hurra geschafft!

Infrastruktur Projekt – RAC (2-4 Knoten)

- Planungsphase
 - ½ bis 1 Jahr
- Beschaffungsphase
 - ¼ Jahr
- Implementations- & Migrationsphase
 - ¼ bis ½ Jahr
- Budget
 - Zwei schicke Doppelhaushälften in guter Lage
 - Wenn man ehrlich ist, könnten die Häuser auch freistehend sein!

Und irgendwie fühlt sich
das noch nicht so richtig
gut an!

Anwendung(sentwickler) vergessen?

Das Projekt sollte nicht mit
der Bereitstellung der
Infrastruktur enden!

Load Balancing & Failover

■ Load Balancing & Failover

Real Application Clusters

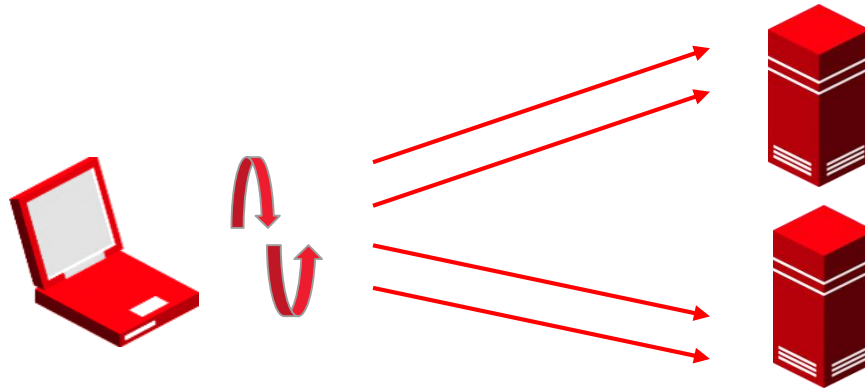
- Ein RAC hat viele Vorteile
 - Lastverteilung
 - Failover
 - Skalierbarkeit
- Leider nicht umsonst
 - Geschickte Konfiguration ist notwendig
 - Viele Fehler möglich (Komplexität)
 - Anwendungen müssen angepasst werden

■ Verbindung zu einer DB/Instanz



```
ORCL.FOO.BAR =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP) (HOST = foo.bar) (PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = ORCL.FOO.BAR)  
    )  
  )  
)
```


■ Client Side Load Balancing

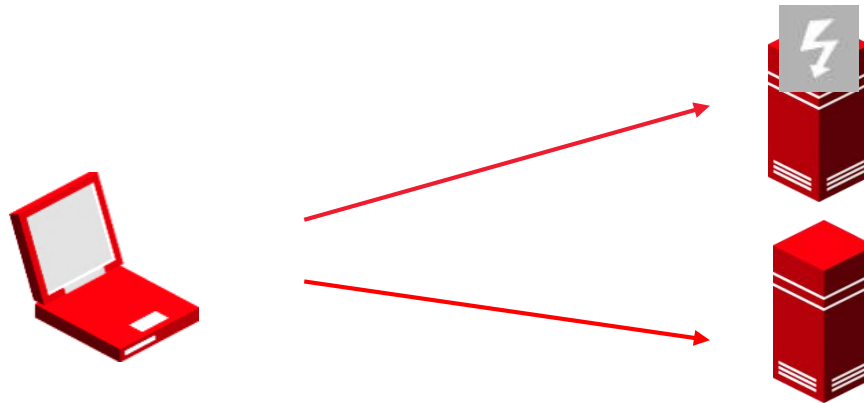


```
RACDB.FOO.BAR =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (LOAD_BALANCE=ON)  
      (ADDRESS = (PROTOCOL = TCP) (HOST = foo1-vip.foo.bar) (PORT = 1521))  
      (ADDRESS = (PROTOCOL = TCP) (HOST = foo2-vip.foo.bar) (PORT = 1521))  
    )  
    (CONNECT_DATA = (SERVICE_NAME = ORCL.FOO.BAR))  
  )  
)
```

■ Client Side Load Balancing

- Einfachste Form der Lastverteilung
- Round Robin
- Möglicherweise keine Informationen zur Auslastung der Knoten
 - Neue Verbindungen können auf bereits ausgelastete Knoten erfolgen
 - Remote Listener Parameter beachten!
- Mit SCAN wird die Auslastung berücksichtigt.
 - Weiterleitung vom SCAN Listener zum Local Listener der Instanz mit der geringsten Auslastung

■ Connect Time Failover

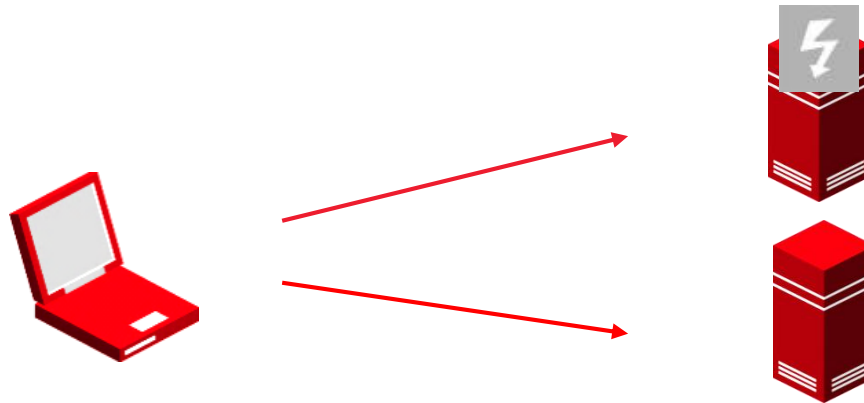


```
RACDB.FOO.BAR =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (LOAD_BALANCE=ON)  
      (FAILOVER = ON)  
      (ADDRESS = (PROTOCOL = TCP) (HOST = foo1-vip.foo.bar) (PORT = 1521))  
      (ADDRESS = (PROTOCOL = TCP) (HOST = foo2-vip.foo.bar) (PORT = 1521))  
    )  
    (CONNECT_DATA = (SERVICE_NAME = ORCL.FOO.BAR))  
  )  
)
```

■ Connect Time Failover

- Instanz ist beim Verbindungsaufbau nicht erreichbar
 - Liste wird sequentiell abgearbeitet
 - Bereits existierende Verbindungen werden nicht neu verbunden
- Failover erfolgt nur beim Verbindungsaufbau
- TCP Timeout kann bis zu zwei Stunden dauern
 - `SQLNET.ORA: sqlnet.outbound_connect_timeout = 1 (s)`

■ Transparent Application Failover - TAF



```
...  
(CONNECT_DATA =  
  (SERVICE_NAME = ORCL.FOO.BAR )  
  (FAILOVER_MODE =  
    ( TYPE = SELECT )  
    ( METHOD = BASIC )  
    ( RETRIES = 3600 )  
    ( DELAY = 1 )  
  )  
)  
...
```

■ Transparent Application Failover - TAF

- Erlaubt ein automatisches Failover bei einem Session Fehler
- Implementiert im OCI Layer des Oracle Clients
- Für RAC-, DG-, FO-Cluster möglich
- Auf Session und Select Level möglich
 - (NONE | SESSION | SELECT)
- Failover-Status kann mittels Event Handler abgefragt werden
- Kann auf Server-Seite und Client-Seite implementiert werden

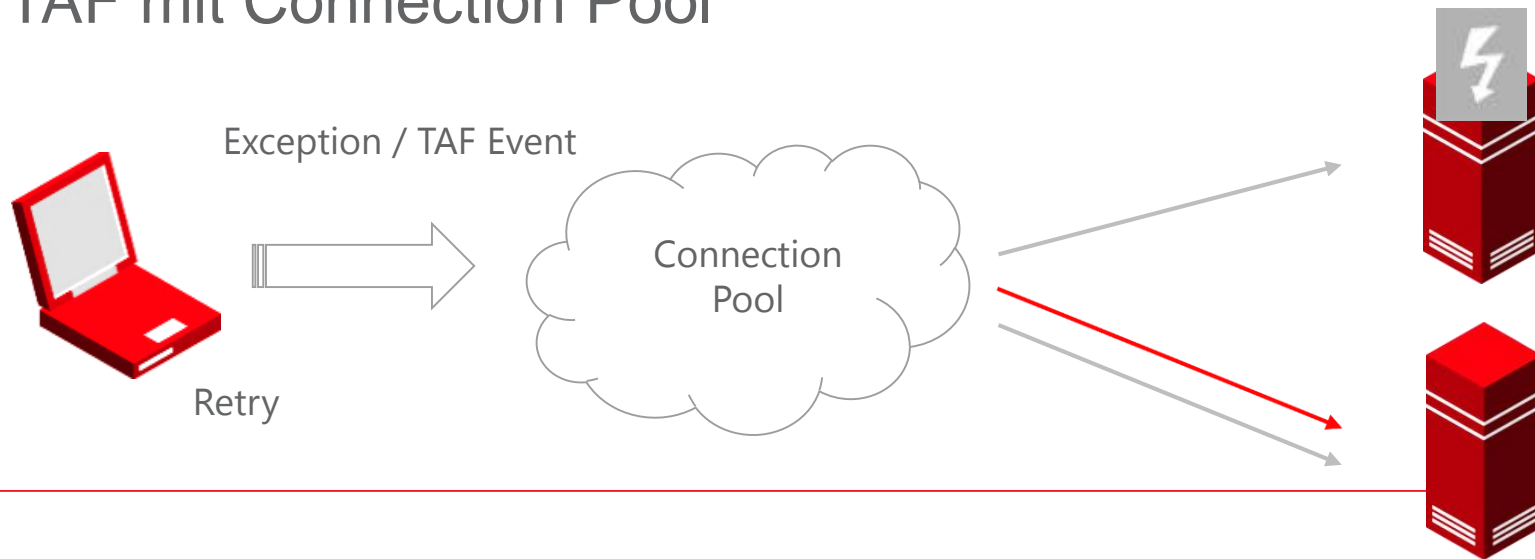
■ TAF – Einschränkungen

- DB Transaktionen können nicht fortgeführt werden
 - Rollback
 - Try/Catch
- Business-Transaktionen müssen überprüft und nachgehalten werden
- Session Parameter müssen ggf. neu gesetzt werden
- TAF kann nicht mit JDBC Thin eingesetzt werden
- Nicht für DB Links möglich

■ TAF Details

- **TYPE = SESSION**
 - SQL Statement schlägt fehl, TAF initiiert Reconnect, Applikation muss Query neu ausführen
- **TYPE = SELECT**
 - OCI lässt das SQL Statement im Hintergrund, Client erhält die fehlenden ROWs, für den Client sieht es nur so aus, als ob er lange wartet, diese funktioniert nicht mit einer offenen Transaktion (Fehler “You need to rollback”)
- **DELAY** und **RETRIES** definieren das Intervall und die max. Anzahl der Verbindungsversuche des OCI-Clients
- **METHOD = BASIC** keine vorbelegte Failover- Sessions/Verbindungen
- **METHOD = PRECONNECT** Vorbelegung der Failover-Verbindung(en)

■ TAF mit Connection Pool



```
...  
(CONNECT_DATA =  
  (SERVICE_NAME = ORCL.FOO.BAR )  
  (FAILOVER_MODE =  
    ( TYPE = SELECT )  
    ( METHOD = BASIC )  
    ( RETRIES = 3600 )  
    ( DELAY = 1 )  
    ( ENABLE = BROKEN )  
  )  
...  
)
```

■ TAF mit Connection Pool

- Kein Feedback von DB-Seite an den Pool
- Fehlerhafte Verbindungen werden nicht automatisch aus dem Pool entfernt
- Dummy Select oder (`ENABLE = BROKEN`) hilft bei der Erkennung von fehlerhaften Verbindungen
- Exception sollten gefangen werden (try/catch).
- Versuchen Sie es doch einfach nochmal

■ Dummy Select für Thin JDBC

Tomcat mit Connection Pool:

```
<Resource name="jdbc/myDatabase" auth="Container"
  type="javax.sql.DataSource"
  driverClassName="oracle.jdbc.OracleDriver"
  url="jdbc:oracle:thin:@
  (DESCRIPTION=(ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=TCP) (HOST=10.0.0.1) (PORT=1521))
    (ADDRESS=(PROTOCOL=TCP) (HOST=10.0.0.2) (PORT=1521))
    (LOAD_BALANCE=yes)) (CONNECT_DATA=(SERVER=DEDICATED)
    (SERVICE_NAME=orcl_jdbc))) "
  username="SCOTT" password="TIGER" maxActive="100" maxIdle="10"
  maxWait="-1" validationQuery="SELECT 1 FROM DUAL"
  testOnBorrow="true"/>
```

■ JDBC Connection Failure Notification

■ JDBC Standalone Client

- Registrierung ONS
- Verwendet FAN Events
 - Node Down

System Property:

```
java -Doracle.jdbc.ConnectionFailureNotification=true
```

Code:

```
Properties props = new Properties();  
props.put("ConnectionFailureNotification", "true");  
Connection = DriverManager.getConnection(url, props);
```

ONS Konfiguration:

```
props.setONSConfiguration("nodes=foo1:6200,foo2:6200");
```

DB Services

■ DB Service

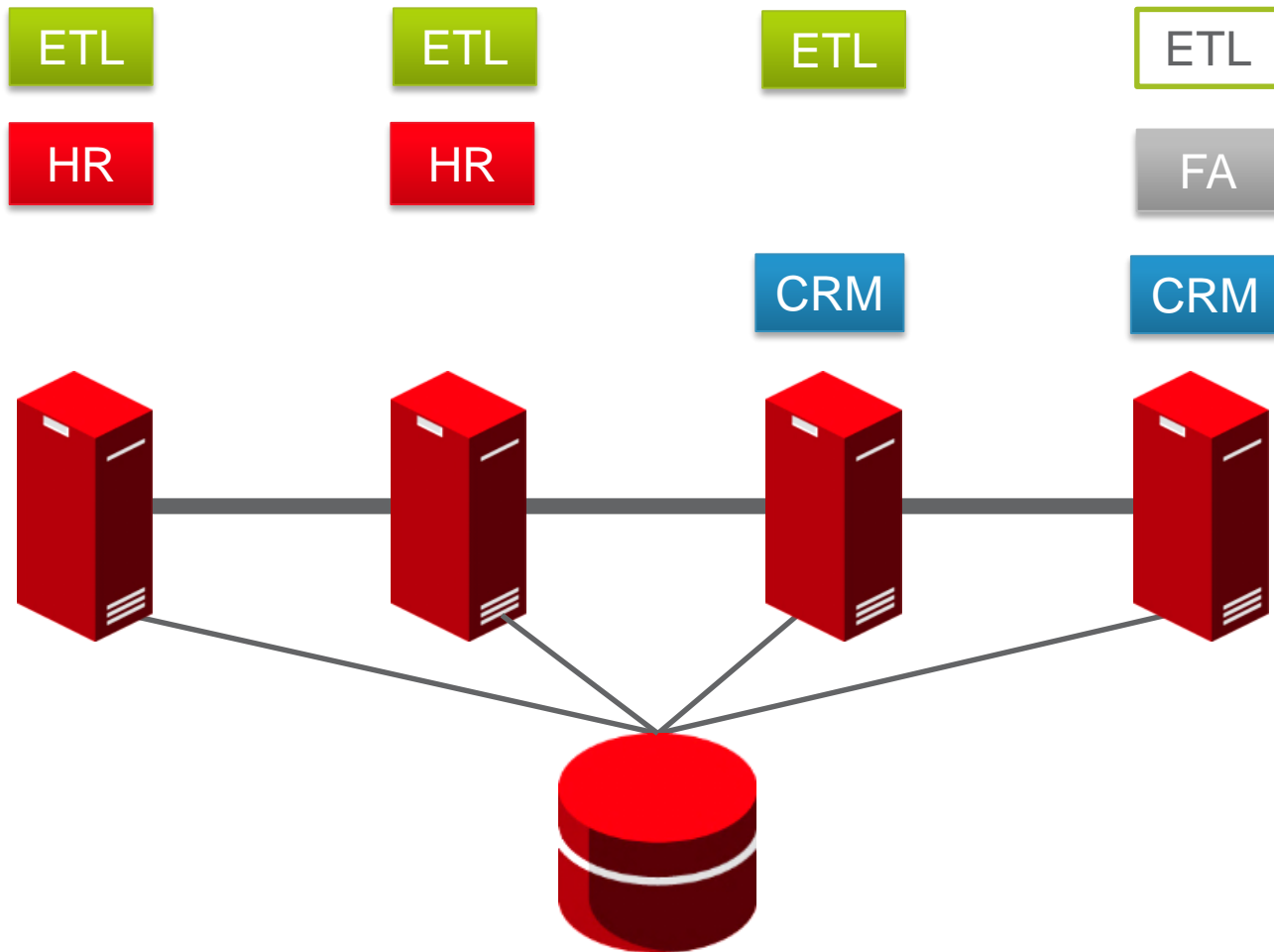
- Wesentlicher Bestandteil von Oracle Net und DB
- Ein Service kann auf mehrere Instanzen zeigen
- Sessions können zweckgebunden gruppiert werden
 - Applikationen
 - Client Verbindungen
 - DBA Zugriff
- Serverseitige Konfiguration für
 - Load Balancing
 - Failover Methode (TAF)
 - FAN Events (AQ)

```
SQL> select name from v$active_services;  
  
NAME  
-----  
ORCL.FOO.BAR  
admin.foo.bar  
client.foo.bar  
SYS$BACKGROUND  
SYS$USERS
```

■ DB Service

- Vereinfachte Client-Administration
- Unmittelbar aktiv
- DBMS_SERVICE-Package kann für die Konfiguration verwendet werden
 - Ab 11gR2 (RAC/HAS) nur noch mittels `SRVCTL`
- Überschreibt Client-Settings und schützt vor Fehlern
- View `DBA_SERVICES` für Service Parameter
- Aktive Services sind im `gv$active_services` VIEW sichtbar

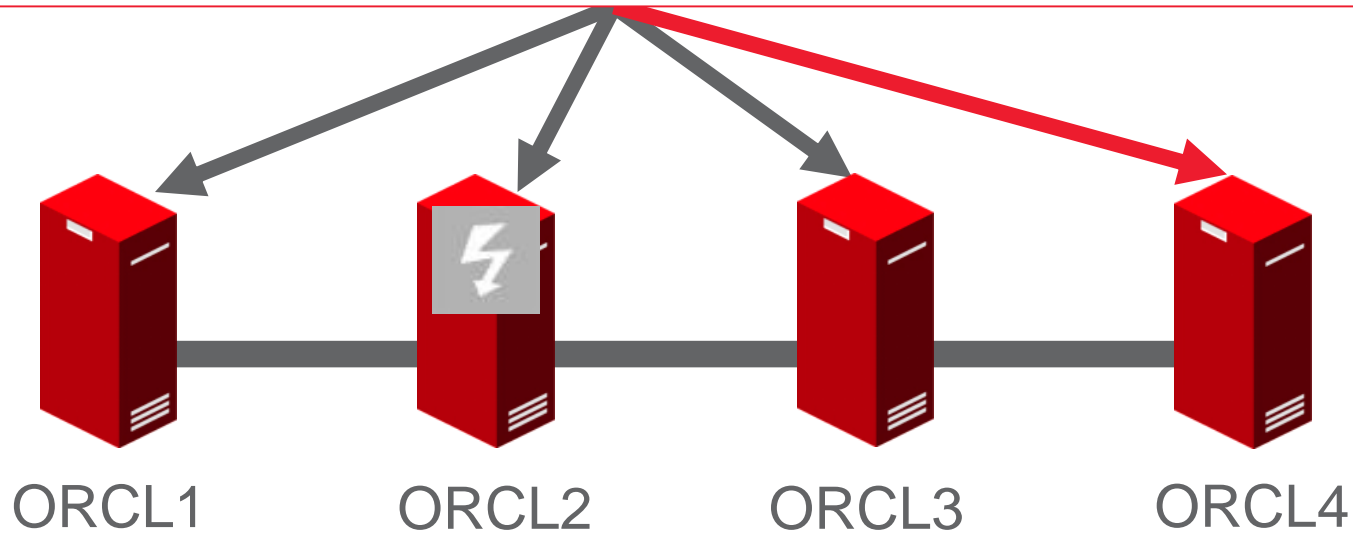
■ Service – Skalierbarkeit



■ Service Anlegen

Beispiel mit SRVCTL:

```
>srvctl add service -d ORCL -s etl.foo.bar \  
-r ORCL1,ORCL2,ORCL3 \  
-a ORCL4
```



■ Service Anlegen – Server Side TAF

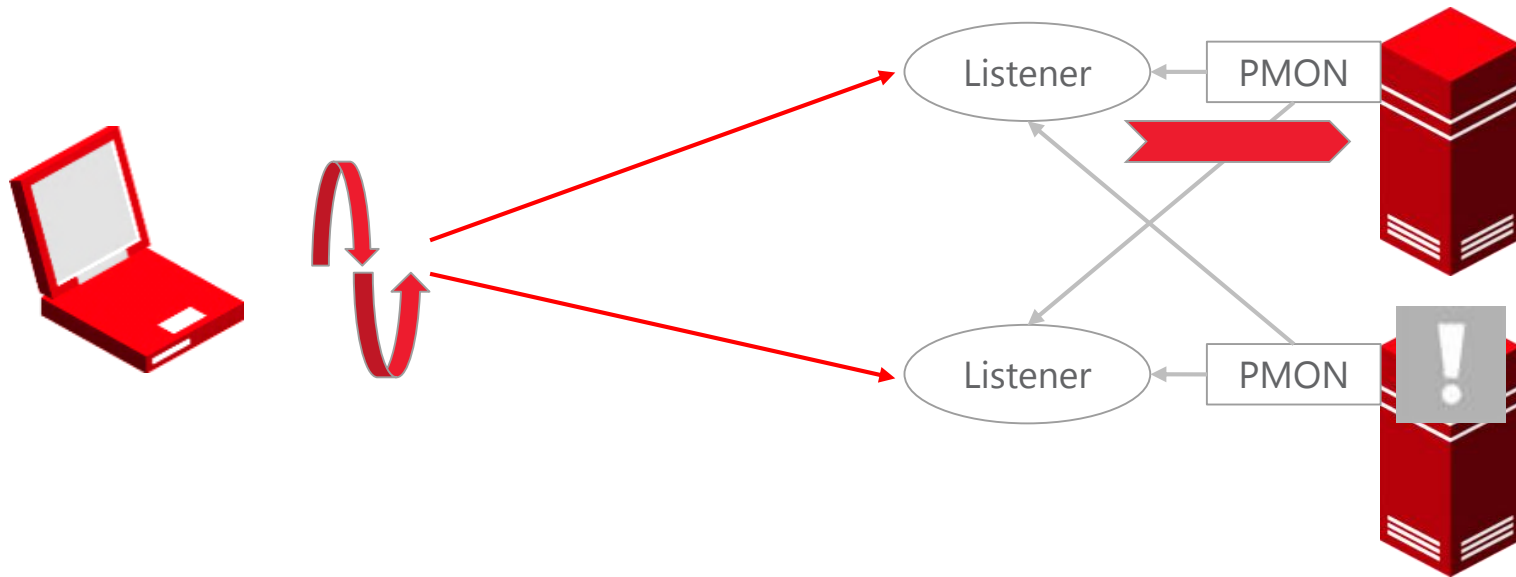
Beispiel mit srvctl:

```
>srvctl add service -d orcl -s orcl_taf.foo.bar -r ORCL1, ORCL2 \  
-P BASIC -e SELECT -m BASIC -z 3600 -w 1
```

```
select  
  schemaname, failover_type, failover_method, failed_over  
from gv$session  
where service_name = 'orcl_taf.foo.bar';
```

SCHEMANAME	FAILOVER_TYPE	FAILOVER_METHOD	FAILED_OVER
SCOTT	SELECT	BASIC	NO
SCOTT	SELECT	BASIC	NO
SCOTT	SELECT	BASIC	NO
SCOTT	SELECT	BASIC	NO

■ Server Side Load Balancing



```
SQL>alter system set remote_listener=  
' (ADDRESS_LIST=  
(ADDRESS=(PROTOCOL=TCP) (HOST=foo2-vip.foo.bar) (PORT=1521)) '  
scope=spfile sid='ORCL1';
```

■ Server Side Load Balancing

- Connect Time Load Balancing
- Wird über DB Services parametrier
- Veränderte Bedingungen werden nicht berücksichtigt
- DB Service Parameter `CLB_GOAL`
 - LONG (default)
 - Sessions werden möglichst gleichmäßig verteilt
 - Session basierte Statistiken
 - Ideal für homogene Aufgaben
 - SHORT
 - Neue Sessions werden an die Instanz mit der geringsten Auslastung geleitet
 - CPU basierte Statistiken
 - Ideal für heterogene Aufgaben und kurze Verbindungen

■ Runtime Connection Load Balancing

- Load Balancing Advisory Goal (GOAL)
 - Unterscheidung nach Reaktionszeit und max. möglichem Durchsatz
 - Service Time
 - Basierend auf Zeit (Dauer) und verfügbarer I/O-Bandbreite
 - Gut für heterogene Verbindungen mit unterschiedlichen Laufzeiten
 - OLTP
 - Troughput
 - Schnellst mögliche Verarbeitung pro Task
 - Basiert auf Prozessorzeit und der höchsten I/O-Leistungsfähigkeit
 - Gut für DWH und Batch Verarbeitungen

■ Runtime Connection Load Balancing

Beispiel einer Service Konfiguration mit der Service Time-Metrik für OLTP und dem `CLB_GOAL=LONG` für langfristige Verbindungen mit einem Connection Pool.

```
srvctl add service -d orcl -s orcl_rlb.foo.bar -r ORCL1,ORCL2 \  
-B SERVICE_TIME -j LONG
```

■ DBA_SERVICES

```
desc dba_services;
Name                               Null Type
-----
...
FAILOVER_METHOD                    VARCHAR2 (64)
FAILOVER_TYPE                      VARCHAR2 (64)
FAILOVER_RETRIES                   NUMBER (10)
FAILOVER_DELAY                     NUMBER (10)
MIN_CARDINALITY                    NUMBER
MAX_CARDINALITY                    NUMBER
GOAL                              VARCHAR2 (12)
DTP                                VARCHAR2 (1)
ENABLED                            VARCHAR2 (3)
AQ_HA_NOTIFICATIONS              VARCHAR2 (3)
CLB_GOAL                          VARCHAR2 (5)
...
```

Fast Application Notification & Fast Connection Failover

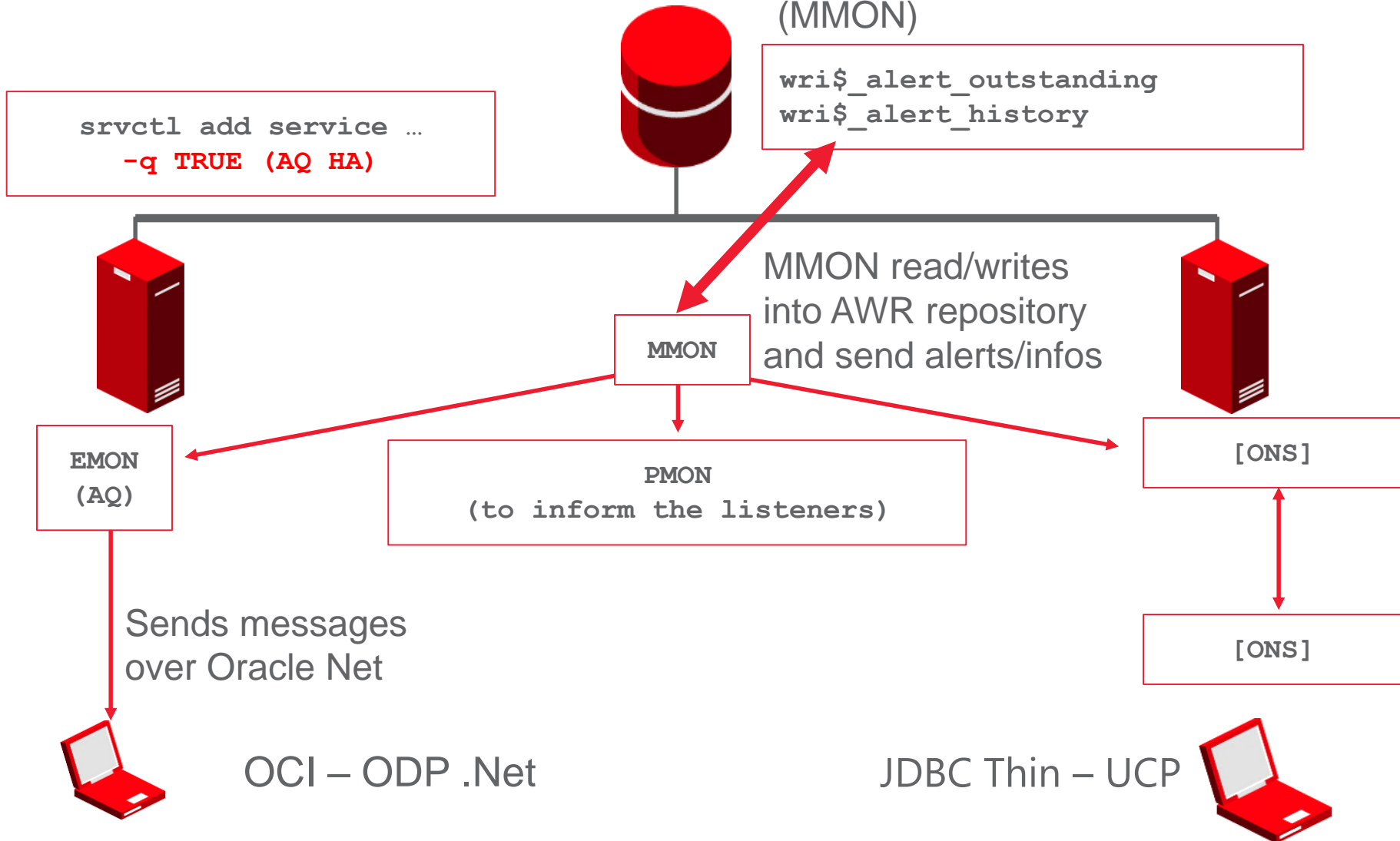
■ Fast Application Notification - FAN

- Informiert Clients...
 - User
 - Connection Pools
 - Applikationsserver
- ...über...
 - Auslastung
 - Störungen
 - Node up/down
 - Instance up/down
 - Database up/down
 - Failover Status
 - Failing over
 - Failed over
 - isValid



FAN Events – Architecture

Automatic Database Diagnostic Monitor writes/uses the Advanced Workload Repository (MMON)



■ FAN Events

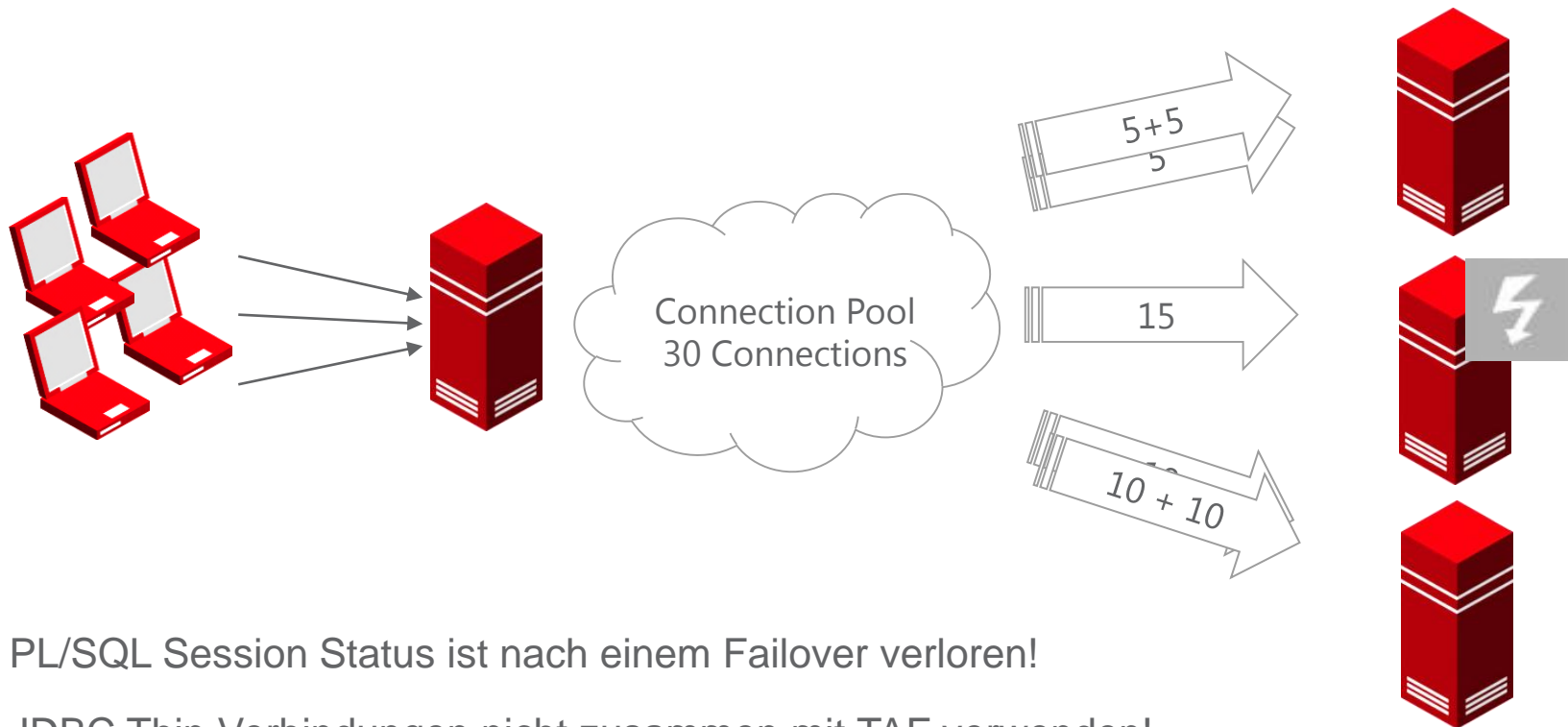
- FAN Events werden mittels ... bereitgestellt
 - Advanced Queueing - AQ
 - Oracle Notification Service – ÓNS
- Clients können sich registrieren und die Events auslesen
 - FAN Events stellen nur Daten zur Verfügung
 - Die Verarbeitung muss von den Clients ausgehen

■ Fast Connection Failover - FCF

- Client Implementation um FAN Events zu verarbeiten
- Nur in Verbindung mit Oracle RAC oder Oracle Restart/HAS und **Connection Pool** möglich
- Standard Lösung für .Net und JAVA verfügbar
- Automatische Reaktion auf Fehler und veränderte Auslastung.
 - Inaktive Verbindungen werden neu sortiert (LOAD)
 - Gestörte Verbindungen werden geschlossen und neu verteilt.
- Verbindungsstatus kann abgefragt werden (`isValid`)

Fast Connection Failover

Nach einem Fehler werden die Verbindungen neu Verteilt



PL/SQL Session Status ist nach einem Failover verloren!

JDBC Thin Verbindungen nicht zusammen mit TAF verwenden!

Oracle Data Provider & Universal Connection Pool

■ Oracle Data Provider for .Net – ODP .Net

- Basiert auf dem Oracle Call Interface – OCI
- Speziell für .Net Applikationen
- FAN Events via Advanced Queuing
- Optimierter ADO .NET DB Zugriff
- FCF fähiger Connection Pool
- Sehr gute Integration in Visual Studio
 - Oracle Data Access Components – ODAC

■ Vorbereitung ODP - DB

- DB-Parameter AQ_TM_PROCESSES >= 1
- Service muss AQ_HA_NOTIFICATIONS aktiviert haben
 - Wird über den Service gesetzt

```
srvctl add service -d orcl -s orcl_otp -r orcl1,orcl2 \  
                  -P basic -e select -m basic -z 3600 -w 1 \  
                  -q TRUE -B SERVICE_TIME -j LONG
```

- Events werden in Queue-Tabellen verfügbar gemacht
 - ALERT_QT
 - SYS\$SERVICE_METRICS_TAB
- User, die die Queue auslesen wollen, benötigen Berechtigung:

```
EXEC DBMS_AQADM.GRANT_QUEUE_PRIVILEGE('DEQUEUE', \  
                                       'SYS.SYS$SERVICE_METRICS', 'SCOTT');
```


■ ODP – FAN Events

Mittels des SELECTs erfolgt die Gegenprobe:

```
select enq_time, user_data from SYS.sys$service_metrics_tab;
...
26.08.13 10:41:44,887738000
SYS.SYS$RLBTYP('orcl_odp.foo.bar', 'VERSION=1.0 database=ORCL
service=orcl_odp.foo.bar {
{instance=ORCL2 percent=38 flag=GOOD aff=TRUE}
{instance=ORCL4 percent=22 flag=GOOD aff=FALSE}
{instance=ORCL3 percent=33 flag=GOOD aff=TRUE}
{instance=ORCL1 percent=9 flag=GOOD aff=TRUE} } timestamp=2013-08-26
12:41:44')
...
```

Die unterschiedlichen Zeitangaben kommen dadurch zustande, dass ENQ_TIME in UTC angegeben ist und timestamp der local time entspricht.

■ ODP – FAN Events

- Verteilung der Sessions (nur Tendenz)
 - Aktive Sessions werden nicht neu verteilt

```
select inst_id, count(*)  
from gv$session where  
service_name = 'orcl_otp.foo.bar'  
group by inst_id order by inst_id;
```

INST_ID	COUNT(*)
1	2
2	18
3	16
4	6

■ ODP – Client Konfiguration

- Connect Time Load Balancing
 - Hot Spot auf den Listenern vermeiden

```
RACDB.FOO.BAR =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (LOAD_BALANCE=ON)
      (ADDRESS = (PROTOCOL = TCP) (HOST = foo1-vip.foo.bar ) (PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP) (HOST = foo2-vip.foo.bar ) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = orcl_odp.foo.bar )
    )
  )
)
```

■ ODP – Client Verbindung

```
// C#
using System;
using Oracle.DataAccess.Client;

class ConnectionSample
{
    static void Main()
    {
        OracleConnection con = new OracleConnection();

        //using connection string attributes to connect to Oracle Database
        con.ConnectionString = "User Id=scott;Password=tiger;Data Source=RACDB;
HA Events=true; Load Balancing=true";
        con.Open();
        Console.WriteLine("Connected to Oracle" + con.ServerVersion);

        // Close and Dispose OracleConnection object
        con.Close();
        con.Dispose();
        Console.WriteLine("Disconnected");
    }
}
```

■ Universal Connection Pool – UCP

- Oracle Standard Lösung für Java Anwendungen
- Verwendet den Oracle JDBC Treiber
- Erhält FAN Event via Oracle Notification Service - ONS
- Keine TAF Unterstützung (JDBC Thin)
- FCF fähiger Connection Pool

■ Oracle Notification Service - ONS

- Ist wesentlicher Bestandteil eines RACs
- Austausch von Status Informationen
- Automatisch als Daemon installiert
- Konfiguration ist dynamisch (ONS-Agent)
- SRVCTL löst RACONS ab

```
>onsctl ping  
ONS is running
```

■ Oracle Notification Service

- Konfiguration wird ab 11gR2 immer in der OCR abgelegt
 - `useocr=true` (vor 11gR2)

```
>cat ORACLE_HOME/opmn/conf/ons.config.  
localport=6100 # line added by Agent  
allowgroup=true  
usesharedinstall=true  
remoteport=6200 # line added by Agent  
nodes=foo1:6200,foo2:6200,foo3:6200,foo4:6200 # line added by Agent
```

- Konfiguration mittels `SRVCTL`

```
srvctl config nodeapps -s  
ONS exists: Local port 6100, remote port 6200, EM port 2016
```

■ Vorbereitung UCP - DB

- Service Konfiguration
 - Kein TAF, keine HA_AQ Services

```
>srvctl add service -d orcl -s orcl_jdbc -r orcl1,orcl2  
-B SERVICE_TIME -j SHORT
```

```
select name from dba_services;
```

```
NAME
```

```
-----
```

```
SYS$BACKGROUND
```

```
SYS$USERS
```

```
ORCL
```

```
orcl_jdbc
```

- Firewall? Ports für den ONS Service freigegeben?

■ UCP – Client Konfiguration

- ONS (zwei Möglichkeiten)
 - Auf Client-Seite eigenen ONS-Daemon installieren
 - Remote Konfiguration (bevorzugt)
- Benötigte Dateien
 - Ucp.jar
 - Ojdbc6.jar
 - Ons.jar
- Müssen via `Classpath Variable` referenziert werden
- `Oracle.ons.oraclehome` System Property muss `ORACLE_HOME` entsprechen
 - `java -Doracle.ons.oraclehome=$ORACLE_HOME ...`

■ UCP Initialisieren

```
PoolDataSource pds = PoolDataSourceFactory.getPoolDataSource();
pds.setConnectionPoolName("fcf_pool");
pds.setFastConnectionFailoverEnabled(true);
pds.setONSConfiguration("nodes=foo1:6200,foo2:6200\nwalletfile=
/oracle11/onswalletfile");
pds.setConnectionFactoryClassName("oracle.jdbc.pool.OracleDataSource");
pds.setURL("jdbc:oracle:thin@(DESCRIPTION= "+
"(LOAD_BALANCE=on) "+
"(ADDRESS=(PROTOCOL=TCP) (HOST=foo1-vip) (PORT=1521)) "+
"(ADDRESS=(PROTOCOL=TCP) (HOST=foo2-vip) (PORT=1521)) "+
"(CONNECT_DATA=(SERVICE_NAME=orcl_jdbc)))");
```

FCF einschalten

```
pds.setFastConnectionFailoverEnabled(true);
```

**Gute Demo: Oracle® Database Express Edition 2 Day + Java
Developer's Guide 11g Release 2**

Zusammenfassung

■ Zusammenfassung

- Wir haben Merkmale der HA-Technologien kennengelernt:
 - Failover
 - Connection Time Failover
 - Transparent Application Failover - TAF
 - Fast Connection Failover - FCF
 - Load Balancing
 - Client-Side Connect Time Load Balancing (Round Robin)
 - Server-Side Connect Time Load Balancing
 - Real Time Connection Load Balancing

■ Zusammenfassung

- Wir wissen wie wir an die FAN Events kommen können:
 - ODP .NET/OCI
 - Advanced Queueing
 - Service Konfiguration – HA Events
 - ALERT_QT
 - SYS\$SERVICE_METRICS_TAB
 - JDBC
 - Oracle Notification Service – ONS
 - Client-Side ONS Daemon
 - Remote Konfiguration

■ Zusammenfassung

- Wir wissen dass aktive Transaktionen im Fehlerfall nicht wiederholt werden können und die Businesslogik überprüft werden muss.
- Damit das gelingt sollten DBAs mit den Anwendungsentwicklern darüber reden.
 - DB Services sollten Applikationsspezifisch bereit gestellt werden
 - Load Balancing Alogarismus
 - Aktive Nodes / Skalierung
 - Failover Methode(n)
 - Execptions sollten Business Transaktionssicherheit gewährleisten

Dann macht es auch Spaß

■ Ausblick

Oracle 12c:

- Application Continuity & Transaktion Guard
 - Aufzeichnen des Status der aktiven Transaktionen über einen Verbindungsverlust hinweg
 - Oracle Exception - IsRecoverable Property
 - FALSE: Roll-Back, Re-Execute und Resubmit
 - TRUE: Logical Transaction Identifiers (LTXID)
 - Entsprechende Maßnahmen ableiten (commit)
- ODP .Net
 - Verwendet nun ONS und nicht mehr Advanced Queuing für die FAN Events
 - ONS ist in Memory basiert und daher schneller als eine Advanced Queue
 - Konsolidierung von JDBC (Thin) und ODP .Net/OCI

Zurücklehnen...

tief Durchatmen!

Das Thema wird komplexer.

Versprochen!

To be continued...



Fragen und Antworten...

Martin Schmitter

Senior Consultant

Tel.: +49-211-58 66 64 70

Martin.schmitter@trivadis.com



BASEL BERN BRUGG LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

2013 © Trivadis

Connecting and using RAC
21.11.2013

trivadis
makes IT easier. ■ ■ ■

Trivadis an der DOAG

Ebene 3 - gleich neben der Rolltreppe

Wir freuen uns auf Ihren Besuch.

Denn mit Trivadis gewinnen Sie immer.

■ Quellen

- Oracle Data Provider for .Net
 - <http://www.oracle.com/technetwork/topics/dotnet/index-085163.html>
- JDBC
 - Oracle® Universal Connection Pool for JDBC Developer's Guide 11g Release 2 (11.2)
 - Oracle® Database JDBC Developer's Guide 11g Release 2 (11.2)
- Oracle Net
 - Oracle® Database Net Services Administrator's Guide 11g Release 2 (11.2)
- RAC
 - Oracle® Real Application Clusters Administration and Deployment Guide 11g Release 2 (11.2)