

Oracle9i RAC

og

Transparent Application Failover
(TAF)

Ingemar Jansson Haverstad

ingemar@oraklet.no

Mine tjenester

- Rådgivning og tilrettelegging med fokus på *Oracle9i*:
 - Avansert replikering
 - Distribuerte databaser
 - Datavaruhus
 - Skalering / Parallellisering
 - Real Application Cluster

Agenda

- Transparent Application Failover
- Oppsett
- Bruk av Transparent Application Failover
- Demo

Transparent Application Failover

- Transparent Application Failover:

Aruntime failover for high availability environments, such as Real Application Clusters. TAF refers to the failover and reestablishment of application-to-service connections. It enables client applications to automatically reconnect to the database if the connection fails, and optionally resume a SELECT statement that was in progress. This reconnect happens automatically from within the Oracle Call Interface (OCI) library.

- Transparent - *Adj. gjennom siktig*
- Translucent - *Adj. gjennom skinnelig*
- Failover
 - Arbeide videre, uten avbrudd...
 - Automatisk oppkobling

Vesentlige faktorer...

- Klient / Server database oppkoblinger.
- Brukerens sessjoner hvor kommandoer utføres.
- Åpne *cursors* som brukes for å hente data.
- Aktive transaksjoner.
- Program variabler på server-siden.

Hvor kan TAF benyttes?

- Planlagte stopp
- Kontrollert vedlikehold
- Last balansering
- Klient aktivitet under en overtakelse

Planlagte stopp

- Av og til er det behov for planlagte stopp. Da er det mulig å bruke TRANSACTIONAL ved stopp på en instans.

SHUTDOWN TRANSACTIONAL

–Stopp blir forsinket til alle aktive transaksjoner er avsluttet.

SHUTDOWN TRANSACTIONAL LOCAL

–Forhindrer nye transaksjoner på denne instansen. Utfør en direkte stopp etter at alle aktive transaksjoner er avsluttet.

Lastbalansering

- Last balansering er mulig i både *shared* og *dedikert* modus.
 - **Lastbalansering for *Shared Server*:**
 - 1) Minst belastet node
 - 2) Minst belastet instans
 - 3) Minst belastet *dispatcher* for den instansen
 - **Lastbalansering for *Dedikert Server*:**
 - 1) Minst belastet node
 - 2) Minst belastet instans
- Kan justere for lastbalansering mellom antallet noder også. Listener filer oppdateres automatisk.
 - 2 noder -> 3 noder -> 4 noder -> 3 noder

Avbrudd ved spørring

- Klienten skjermes mest mulig ved avbrudd.
- Hvis data er tilgjengelig i buffer cachen på den andre noden er nedetiden kort.
- Ved en krasj gjenoppretter de overlevende nodene informasjonen från den noden som krasjer. Dette kan medføre en lengre pause.
- Kan benytte PRECONNECT for å ha en tilgjengelig forbindelse på den overtagende noden. Gir kortere responstid men bruker ressurser i forkant.

Avbrudd ved *DML*

- *DML* eller *Data Manipulation Language* klienter utfører *INSERT*, *UPDATE* og *DELETE*.
- Oracle håndterer enkelte feil og kobler opp klienten igjen.
- Klienten vil oppleve en (kort) heng fordi de gjenværende nodene gjenoppretter bekreftede transaksjoner.
- Skriver kode i applikasjonen som fortsetter med arbeidet på en hensiktsmessig måte.

Restriksjoner

- *PL/SQL* pakke statuser mistes ved en overtakelse.
- `ALTER SESSION` setninger mistes.
- Aktive transaksjoner må rulles tilbake og utføres på nytt:
 - **OCI:** `OCITransRollback` -> suksess -> gjenta
- `TYPE=SELECT` i service som benyttes, spørring fortsetter på overlevende instanse.
- Fortsatt arbeide på en *cursor* kan resultere i en feilmelding.

Oracle Call Interface - OCI

- Fordi gjenoppkobling skjer automatisk innenfor OCI-biblioteker på klienten, må Oracle *OCI* biblioteker benyttes.
- Benytt *JDBC OCI* i stedet for *PL/SQL* pakker.
- Kan ikke bruke tynn JDBC hvis TAF funksjonalitet ønskes.

Java program eksempel

- Eksempel *OCIFailOver.java* i *\$ORACLE_HOME/jdbc/demo/samples/jdbcoci*

```
// TAF callback function
public int callbackFn (Connection conn, Object ctxt, int type, int
event) {
...
switch (event) {
case FO_BEGIN:
    System.out.println(ctxt + ": " + ftype + " failing over...");
    break;
case FO_END:
    System.out.println(ctxt + ": failover ended");
    break;
```

```
William Gietz
null: SELECT failing over...
null: failover error gotten. Sleeping...
null: failover ended
Steven King
```

Failover parametere

- Settes i *tnsnames.ora*

```
db.us.acme.com=
  (description=
    (load_balance=on) /* only connect time load balancing and */
                        /* connection load balancing */
    (failover=on)      /* only connect time failover */
  (address=
    (protocol=tcp)
    (host=db1-server)
    (port=1521))
  (address=
    (protocol=tcp)
    (host=db2-server)
    (port=1521))
  (connect_data=
    (service_name=db.us.acme.com)
    (failover_mode=
      (type=select)
      (method=basic)))
```

FAILOVER_MODE

- Settes i *tnsnames.ora* i *CONNECT_DATA* delen

```
...  
(connect_data=  
  (service_name=db.us.acme.com)  
    (failover_mode=  
      (type=select)  
      (method=basic))))
```

- **FAILOVER_MODE**

- **BACKUP** backup-server for PRECONNECT
- **TYPE** BASIC/SELECT
- **METHOD** BASIC/PRECONNECT
- **RETRIES** antall ganger oppkoblings skal prøves
- **DELAY** ventetid i sekunder innen nytt forsøk

Sjekk oppkoblingen...

- Bruk *tnsping* eller se i nett-loggen *listener.log*

```
$ tns ping TEST
```

```
TNS Ping Utility for Linux: Version 9.2.0.1.0 -  
Production on 20-MAY-2003 13:48:14
```

```
Used parameter files:
```

```
/oracle/product/ora920/network/admin/sqlnet.ora
```

```
Used TNSNAMES adapter to resolve the alias
```

```
Attempting to contact (DESCRIPTION = (FAILOVER = ON)
```

```
(ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = tuppen)
```

```
(PORT = 1521))) (CONNECT_DATA = (SERVER = DEDICATED)
```

```
(SERVICE_NAME = TEST.oracle.no) (FAILOVER_MODE = (TYPE =
```

```
SELECT) (METHOD = BASIC))))
```

```
OK (10 msec)
```

Bekreftelsen

- Tre kolonner i *V\$SESSION* kan sjekkes:

```
SELECT
  MACHINE, FAILOVER_TYPE, FAILOVER_METHOD, FAILED_OVER, COUNT(*)
FROM V$SESSION
GROUP BY MACHINE, FAILOVER_TYPE, FAILOVER_METHOD, FAILED_OVER;
```

MACHINE	FAILOVER_TYPE	FAILOVER_M	FAI	COUNT(*)
db1-server	NONE	NONE	NO	11
db2-server	SELECT	PRECONNECT	NO	1

Og etter den simulerte krasjen...

MACHINE	FAILOVER_TYPE	FAILOVER_M	FAI	COUNT(*)
db2-server	NONE	NONE	NO	10
db2-server	SELECT	PRECONNECT	YES	1

Og

nå

DEMO!