

Oracle 10g Real Application Clusters: konfiguracja i administrowanie

Maciej Zakrzewicz
PLOUG
mzakrz@cs.put.poznan.pl

Strzeszczenie

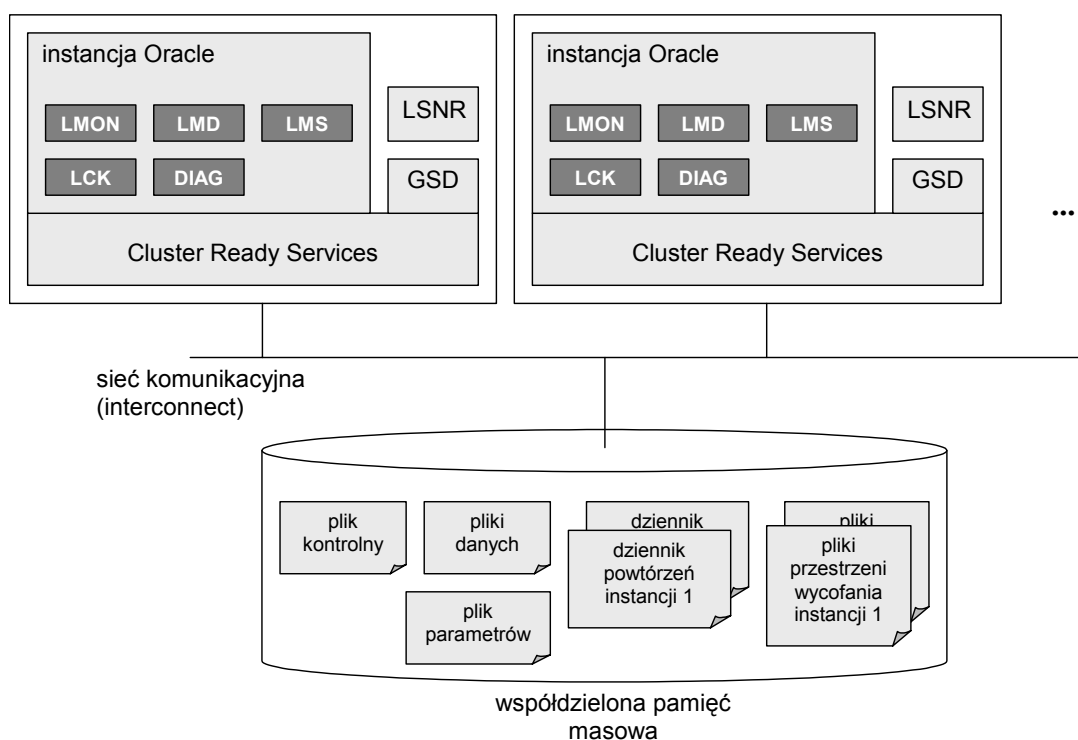
Real Application Clusters (RAC) to dystrybucja serwera bazy danych Oracle 10g przeznaczona do wykorzystywania w środowiskach klastrów obliczeniowych, realizująca koncepcję zwielokrotnienia instancji serwera bazy danych w celu zwiększenia niezawodności i wydajności obsługi zapytań. Artykuł omawia zasady projektowania, instalacji, konfiguracji i administrowania systemami baz danych opartymi na architekturze RAC.

1. Wstęp

Klaster sprzętowy (hardware cluster) to zespół komputerów nazywanych węzłami, połączonych ze sobą szybką siecią komunikacyjną, współdzielących urządzenia pamięci masowej, a użytkownikowi aplikacyjnemu prezentujących się jako jeden silny i niezawodny komputer logiczny. Głównymi celami konstrukcji klastrów sprzętowych są: uzyskanie wysokiej odporności na awarie, skalowalność wydajności względem liczby współbieżnych użytkowników oraz uzyskanie wysokiej mocy przetwarzania dla pojedynczego użytkownika. Dla skutecznego wykorzystania własności klastrów sprzętowych niezbędna jest budowa specjalnie zorientowanego oprogramowania, rozwiązującego m.in. kwestie wykrywania awarii, synchronizacji stanu, szeregowania zadań.

Oracle 10g Real Application Clusters (RAC) jest dystrybucją serwera bazy danych Oracle10g przeznaczoną do instalacji w środowiskach klastrów sprzętowych. Od standardowej edycji serwera bazy danych architektura RAC różni się m.in. obecnością wielu instancji obsługujących tę samą bazę danych, implementacją rozproszonego bufora danych oraz dodatkowymi komponentami monitorującymi i administracyjnymi. Uproszczony model architektury RAC przedstawiono na rys. 1. Na każdym węźle umieszczona jest jedna instancja Oracle wraz z procesem nasłuchu Listener (LSNR). Pliki bazy danych znajdują się na współdzielonym urządzeniu pamięci masowej. Wspólne są pliki kontrolne, parametrów i danych. Każda instancja posiada oddzielne pliki dziennika powtórzeń (tzw. wątek dziennika powtórzeń – Log Thread) i pliki przestrzeni wycofania, lecz są one również umieszczone na wspólnym urządzeniu. Współdzielona pamięć masowa może być realizowana jako: zbiór surowych partycji zarządzanych bezpośrednio przez instancje RAC, jako klastrowy system plików (Cluster File System, np. Oracle Cluster File System dla MS Windows/Linux), lub jako Automatic Storage Management (ASM). Instancje RAC

posiadają dodatkowe procesy drugoplanowe: LMON, LMD, LMS, LCK i DIAG. Proces LMON (Global Enqueue Service Monitor) odpowiada za monitorowanie blokad w obrębie całego systemu i za nadzorowanie pracy procesu LMD. Proces LMD (Global Enqueue Service Daemon) obsługuje żądania pobierania i zwalniania blokad. Proces LMS (Global Cache Service) koordynuje rozproszony dostęp do bufora danych oraz realizuje transfer bloków pomiędzy instancjami w ramach mechanizmu nazywanego Cache Fusion. Proces LCK koordynuje rozproszony dostęp do bufora słownika danych. Proces DIAG (Diagnosability Daemon) odpowiada za rejestrowanie w pliku śladu informacji diagnostycznych dotyczących awarii procesów instancji. Instancja RAC korzysta z usług specjalizowanej warstwy oprogramowania nazywanej Cluster Ready Services (CRS), odpowiadającej za komunikację z systemem operacyjnym klastra i za jego monitorowanie. Każdy węzeł jest dodatkowo wyposażony w proces GSD, który wykonuje zadania administracyjne (np. uruchomienie instancji) na rzecz zdalnych narzędzi, np. SRVCTL.



Rys. 1. Zarys architektury RAC

Środowisko RAC oferuje także interesujące mechanizmy równoległego wykonywania zapytań. W przypadku, gdy zadany przez programistę stopień równoległości zapytania nie może być uzyskany na poziomie instancji RAC odpowiedzialnej za wykonywanie zapytania, wówczas następuje przekazanie fragmentów pracy pozostałym instancjom RAC.

2. Konfiguracja instancji Oracle

2.1. Dodatkowe parametry inicjalizacyjne

Od każdej instancji Oracle uczestniczącej w systemie RAC wymaga się posiadania szczególnej konfiguracji, obejmującej: ustawienia parametrów inicjalizacyjnych, odrębnego wątku dziennika powtórzeń oraz oddzielnej przestrzeni wycofania. Poniżej przedstawiono wymagane dodatkowe parametry inicjalizacyjne:

- CLUSTER_DATABASE – uaktywnia tryb pracy RAC
- THREAD - identyfikuje numer wątku dziennika powtórzeń, z którego korzysta instancja; wartość ta musi być niepowtarzalna w obrębie systemu RAC
- INSTANCE_NUMBER – numer porządkowy instancji; wartość ta musi być niepowtarzalna w obrębie systemu RAC; zwykle identyczny jak THREAD
- UNDO_TABLESPACE – nazwa przestrzeni wycofania wykorzystywanej przez instancję; każda instancja musi korzystać z odrębnej przestrzeni wycofania

Przykład :

```
CLUSTER_DATABASE = TRUE  
THREAD = 1  
INSTANCE_NUMBER = 1  
UNDO_TABLESPACE = UNDO01
```

Ponadto, w środowisku instancji RAC zaleca się powiększenie rozmiaru zbiornika współdzielonego (Shared Pool) o 15% i bufora danych (Buffer Cache) o 10% w stosunku do rozmiarów tych obszarów w instancji pojedynczej.

2.2. Plik parametrów inicjalizacyjnych

Parametry inicjalizacyjne wszystkich instancji RAC są przechowywane w jednym wspólnym pliku parametrów typu SPFILE, umieszczonym na współdzielonym urządzeniu pamięci masowej. Zmiana wartości parametrów inicjalizacyjnych wpływających na pracę pojedynczej instancji RAC odbywa się za pomocą poleceń ALTER SYSTEM zawierających klauzulę SID, np.:

```
ALTER SYSTEM  
SET SHARED_POOL_SIZE=100M SCOPE=SPFILE SID=INST2
```

Powyższe polecenie spowoduje zmianę rozmiaru zbiornika współdzielonego w instancji INST2, niezależnie od tego, która instancja RAC otrzymała to polecenie.

3. Instalacja

Instalowanie Oracle 10g Real Application Clusters przebiega dwuetapowo: w pierwszym kroku instalowany jest CRS, w drugim – serwer bazy danych zawierający komponenty RAC. CRS i RAC muszą być instalowane w odrębnych katalogach Oracle Home. Zarówno CRS, jak i RAC są zwykle instalowane na dyskach lokalnych, natomiast na dyskach współdzielonych umieszczane są: pliki bazy danych, pliki konfiguracyjne, plik wotywny (voting file) wykorzystywany przez CSS (20 MB) oraz plik OCR (100MB).

Wymagania sprzętowe dla instalacji Oracle 10g Real Application Clusters w systemie operacyjnym Linux obejmują: minimum 512 MB RAM, minimum 1GB przestrzeni wymiany (swap space), minimum 400 MB przestrzeni tymczasowej /tmp, minimum 4 GB przestrzeni dyskowej. Każdy węzeł musi posiadać co najmniej dwa adaptory sieciowe: jeden publiczny, obsługujący protokół TCP/IP, oraz jeden prywatny, obsługujący protokół UDP (interconnect) – rekomendowany jest tu Gigabit Ethernet. Nazwy interfejsów związane z adapterami sieciowymi muszą być identyczne na wszystkich węzłach klastra. Adres IP i nazwa DNS każdego węzła, wykorzystywane przez publiczny adapter sieciowy, muszą zostać zarejestrowane w pliku /etc/hosts lub serwerze DNS.

Dla poprawnego działania programu instalacyjnego Oracle Universal Installer wymaga się zdefiniowania tzw. równoważności użytkowników (user equivalence) pomiędzy węzłami klastra. W tym celu, na każdym węźle w pliku /etc/hosts.equiv należy umieścić nazwy wszystkich węzłów klastra.

Kolejnym krokiem przygotowania do instalacji jest wybór rodzaju systemu współdzielonej pamięci masowej: klastrowy system plików (np. OCFS) partycje surowe lub ASM. Gdy wybrany zostanie OCFS, należy pobrać wersję instalacyjną oprogramowania z serwera <http://oss.oracle.com/projects/ocfs/>, zainstalować moduły RPM, uruchomić narzędzie ocfstool, wygenerować plik konfiguracyjny ocfs.conf, przygotować partycje dyskowe, sformatować je i zapewnić automatyczne ładowanie OCFS po każdym restarcie systemu operacyjnego. Gdy wybrana zostanie metoda partycji surowych, należy utworzyć niezbędną liczbę partycji za pomocą narzędzia fdisk, a następnie przyłączyć je do systemu operacyjnego edytując plik konfiguracyjny /etc/sysconfig/rawdevices. Ponadto, aby umożliwić programowi DBCA odnalezienie właściwych partycji, należy utworzyć plik odwzorowujący partycje na pliki bazy danych: nazwabazy_raw.conf.

Po wykonaniu powyższych czynności wstępnych przystępujemy do instalowania oprogramowania CRS, a następnie do instalowania oprogramowania RAC. Oprogramowanie RAC dla całego klastra może być instalowane z poziomu pojedynczego węzła. Narzędzia instalacyjne umożliwiają także utworzenie nowej bazy danych.

4. Narzędzia administracyjne

4.1. Perspektywy GV\$

Administratorzy pojedynczych instancji Oracle często korzystają z dynamicznych perspektyw wydajności (Dynamic Performance Views), dostarczających informacji o aktualnym obciążeniu, stanie i wydajności systemu, np. V\$SESSION, V\$LOG, V\$PROCESS. W systemie RAC, złożonym z wielu instancji, przydatny może być globalny dostęp do tego typu informacji opisujących nie pojedynczą instancję, lecz pełen zbiór instancji RAC. W tym celu przygotowano lustrzany zbiór perspektyw, nazywanych globalnymi dynamicznymi perspektywami wydajności (Global Dynamic Performance Views), udostępniających sumę zawartości dynamicznych perspektyw wydajności dostarczanych przez wszystkie instancje klastra. Nazwy globalnych dynamicznych perspektyw wydajności rozpoczynają się od liter GV\$, np. GV\$SESSION, GV\$LOG, GV\$PROCESS, itd. W porównaniu z tradycyjnymi perspektywami V\$, perspektywy GV\$ posiadają jedną dodatkową kolumnę – INST_ID – opisującą identyfikator instancji RAC, z której pochodzi informacja źródłowa. Dostęp do perspektyw GV\$ jest możliwy z poziomu dowolnej instancji RAC. Uwaga: aby perspektywy GV\$ były dostępne, parametr inicjalizacyjny PARALLEL_MAX_SERVERS musi dla każdej instancji posiadać wartość niezerową. Wynika to ze specyficznej formy równoległych zapytań, na których oparte są definicje tych perspektyw. Gdyby dla jednej z instancji RAC wartość parametru PARALLEL_MAX_SERVERS była zerowa, to instancja ta zostałaby pominięta w treści dostarczanej przez perspektywy GV\$.

4.2. Enterprise Manager

Narzędzie Enterprise Manager oferuje konsolę administracyjną nazwaną Cluster Database, która umożliwia monitorowanie i administrowanie środowiskiem Oracle 10g Real Application Clusters. Z poziomu konsoli Cluster Database możliwy jest dostęp do szczegółowych paneli Cluster Database Instance, umożliwiających administrowanie pojedynczymi instancjami RAC. Dodatkowym narzędziem jest konsola Cluster, która służy do zarządzania pełnym klastrem sprzętowym.

4.3. Srvctl

Narzędzie SRVCTL to skrypt administracyjny, wywoływany z poziomu wiersza poleceń, umożliwiający zarządzanie środowiskiem Oracle 10g Real Application Clusters w trybie znakowym. Poniżej przedstawiono przykłady użycia SRVCTL.

Uruchamianie pojedynczych instancji RAC:

```
srvctl start instance -d BAZA1 -i INST1, INST2
```

Zatrzymywanie pojedynczych instancji RAC:

```
srvctl stop instance -d BAZA1 -i INST1, INST2
```

Uruchomienie kompletnego systemu RAC:

```
srvctl start database -d BAZA1
```

Zatrzymanie kompletnego systemu RAC w trybie Transactional:

```
srvctl stop database -d BAZA1 -o transactional
```

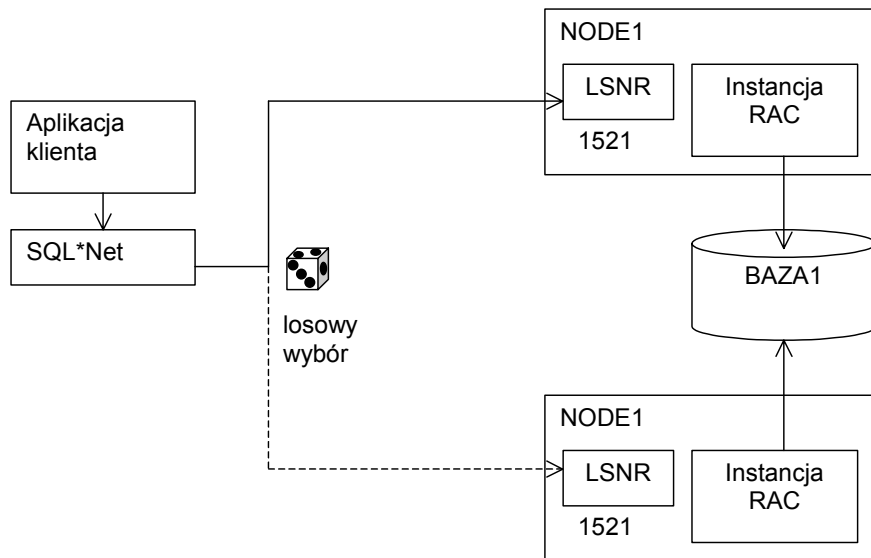
5. Obsługa sytuacji awaryjnych

5.1. SQL*Net: Równoważenie obciążenia w chwili podłączania się użytkownika do bazy danych

Oprogramowanie biblioteki SQL*Net zainstalowanej po stronie aplikacji klienta oferuje mechanizm losowego wyboru jednego z wielu zdefiniowanych procesów nasłuchu (Listener) skojarzonych z instancjami RAC obsługującymi tę samą bazę danych. Dzięki temu, sesje klientów, a tym samym obciążenie, mogą być równomiernie rozpraszane pomiędzy wieloma instancjami RAC. W celu uaktywnienia mechanizmu równoważenia obciążenia biblioteki SQL*Net, konieczne jest umieszczenie zmiennej `LOAD_BALANCE=ON` w pliku konfiguracyjnym klienta sieciowego `TNSNAMES.ORA`, a ponadto związanie wielu alternatywnych adresów (struktura `ADDRESS`) z tą samą docelową usługą bazy danych Oracle. Poniżej przedstawiono fragment przykładowego pliku konfiguracyjnego `TNSNAMES.ORA`:

```
KADRY =
  (DESCRIPTION =
    (LOAD_BALANCE=ON)
    (ADDRESS_LIST =
      (ADDRESS=(PROTOCOL=TCP) (HOST=NODE1) (PORT=1521))
      (ADDRESS=(PROTOCOL=TCP) (HOST=NODE2) (PORT=1521))
    )
    (CONNECT_DATA=(SERVICE_NAME=BAZA1)) )
```

Zauważmy, że z identyfikatorem `KADRY` związane są dwa adresy procesów nasłuchu, znajdujących się na dwóch odrębnych węzłach, jednak pośredniczących w dostępie do tej samej bazy danych `BAZA1`. Powyższa konfiguracja została zilustrowana na rys. 2. W chwili nawiązywania połączenia z bazą danych, biblioteka SQL*Net klienta dokona losowego wyboru jednego z dwóch alternatywnych adresów, a następnie nawiąże połączenie z procesem nasłuchu i docelową instancją RAC.



Rys. 2. SQL*Net: równoważenie obciążenia w chwili podłączania się użytkownika do bazy danych

5.2. Automatyczne ponowienie operacji nawiązywania połączenia w przypadku awarii

Gdy wykorzystywany jest mechanizm równoważenia obciążenia przez SQL*Net, a wylosowany węzeł jest aktualnie niedostępny (awaria), wówczas połączenie nie dochodzi do skutku, a aplikacja użytkownika otrzymuje komunikat o błędzie. W celu umożliwienia automatycznego ponowienia operacji nawiązywania połączenia w przypadku, gdy poprzednia operacja zakończyła się niepowodzeniem, należy uaktywnić mechanizm Failover po stronie biblioteki SQL*Net klienta. Czynność ta wymaga wprowadzenia zmiennej `FAILOVER=TRUE` do pliku `TNSNAMES.ORA`, jak pokazano poniżej:

```

KADRY =
  (DESCRIPTION =
    (LOAD_BALANCE=ON)
    (FAILOVER=ON)
    (ADDRESS_LIST =
      (ADDRESS=(PROTOCOL=TCP) (HOST=NODE1) (PORT=1521))
      (ADDRESS=(PROTOCOL=TCP) (HOST=NODE2) (PORT=1521))
    )
    (CONNECT_DATA=(SERVICE_NAME=BAZA1)))
  
```

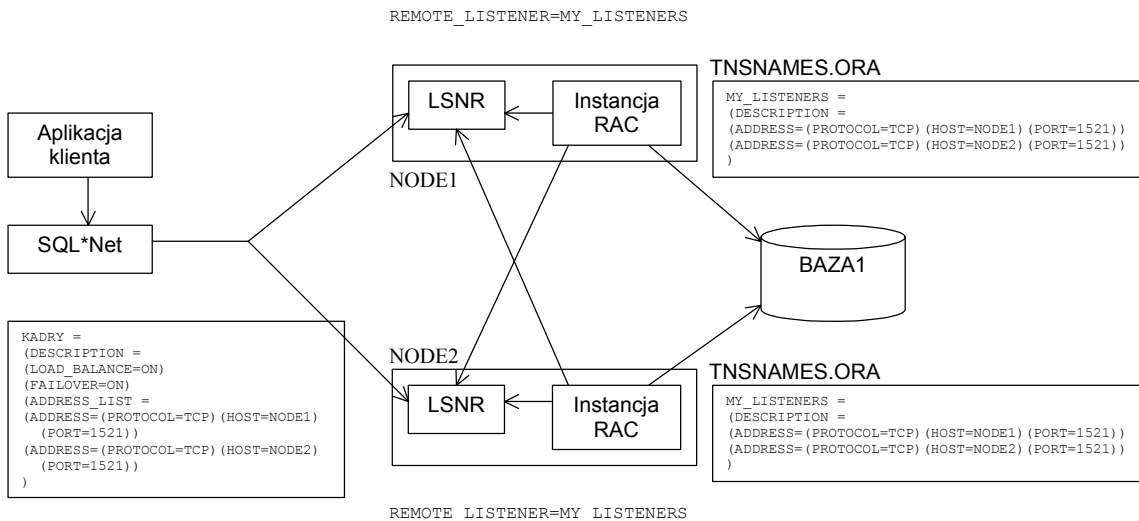
Zgodnie z przedstawioną powyżej konfiguracją, w przypadku gdy SQL*Net losowo wybierze węzeł NODE2, lecz próba nawiązania połączenia nie powiedzie się, wtedy automatycznie zostanie przeprowadzone kolejne losowanie (wśród pozostałych adresów – w omawianym przypadku jest to tylko jeden alternatywny adres) i kolejna próba nawiązania połączenia.

5.3. Listener: Równoważenie obciążenia w chwili podłączania się użytkownika do bazy danych

Funkcje równoważenia obciążenia są implementowane także przez proces nasłuchu sieciowego (Listener). Dzięki mechanizmowi automatycznej rejestracji instancji, każdy proces nasłuchu może posiadać wiedzę o wszystkich dostępnych instancjach RAC oraz podejmować decyzję o przekierowaniu połączenia użytkownika na instancję znajdującą się na innym węźle niż proces nasłuchu, który otrzymał żądanie. Poza wiedzą o dostępności instancji RAC, proces nasłuchu gromadzi również szczegółowe informacje o liczbie nawiązanych sesji oraz o długości kolejki poleceń. Informacje te mogą posłużyć do usprawnienia procedur rozpraszania połączeń pomiędzy węzłami.

W celu uaktywnienia funkcji równoważenia obciążenia przez proces nasłuchu, każda instancja RAC musi wykorzystywać parametr inicjalizacyjny `REMOTE_LISTENER`, którego wartość jest identyfikatorem połączenia zapisanym w pliku `TNSNAMES.ORA`, skojarzonym z definicją procesów nasłuchu (plik `TNSNAMES.ORA` obecny jest tu po stronie serwera). Dzięki temu, uruchamiana instancja RAC dokonuje swojej rejestracji u wszystkich procesów nasłuchu opisanych w pliku `TNSNAMES.ORA`. Proces nasłuchu, który otrzyma od aplikacji użytkownika żądanie nawiązania połączenia z bazą danych, może wówczas podjąć decyzję o przekierowaniu tego połączenia do procesu nasłuchu związanego z najmniej obciążoną instancją RAC pracującą na najmniej obciążonym węźle klastra.

Przykład działania mechanizmu równoważenia obciążenia przez proces nasłuchu przedstawiono na rys. 3. W chwili uruchomienia instancji RAC na węzłach `NODE1` i `NODE2` następuje ich wzajemna rejestracja u procesów nasłuchu. Gdy aplikacja klienta nawiązuje połączenie z bazą danych, wówczas pomimo wcześniejszego wyboru jednego z procesów nasłuchu przez bibliotekę `SQL*Net` klienta, proces ten ma możliwość przekierowania żądania do procesu nasłuchu znajdującego się na innym węźle, o ile węzeł ten cechuje się aktualnie niższym obciążeniem.



Rys. 3. Listener: równoważenie obciążenia w chwili podłączania się użytkownika do bazy danych

5.4. Transparent Application Failover (TAF)

Biblioteka OCI (Oracle Call Interface), wykorzystywana przez programistów aplikacji do komunikacji z SQL*Net, oferuje interesującą funkcję automatycznego wznawiania sesji użytkownika w sytuacji awarii. Gdy awaria węzła lub instancji RAC nastąpi podczas pracy użytkownika, OCI może automatycznie nawiązać nową sesję z inną, sprawną instancją. Aktualna transakcja zostanie utracona (wycofana), aczkolwiek gdyby awaria (i przełączenie) nastąpiła w chwili realizacji długotrwałego polecenia SELECT, wówczas istnieje możliwość automatycznego wznowienia przerwanej polecenia natychmiast po przełączeniu. Mechanizm jest domyślnie nieaktywny. Administrator dokonuje jego aktywacji i konfiguracji przy użyciu pliku konfiguracyjnego TNSNAMES.ORA. Poniżej przedstawiono przykład pliku TNSNAMES.ORA zawierającego parametry konfiguracyjne TAF (sekcja FAILOVER_MODE):

```
KADRY =
  (DESCRIPTION =
    (LOAD_BALANCE=ON)
    (FAILOVER=ON)
    (ADDRESS_LIST =
      (ADDRESS=(PROTOCOL=TCP) (HOST=NODE1) (PORT=1521))
      (ADDRESS=(PROTOCOL=TCP) (HOST=NODE2) (PORT=1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = BAZA1)
      (FAILOVER_MODE =
        (TYPE=SESSION)
        (METHOD=BASIC)
        (RETRIES=100)
        (DELAY=10))))
```

Znaczenie zmiennych użytych w powyższym przykładzie jest następujące. Zmienna TYPE określa, czy po przełączeniu nastąpi rekonstrukcja otwartych kursorów, umożliwiającą kontynuowanie przerwanych poleceń SELECT. Zmienna TYPE przyjmuje jedną z dwóch wartości: SESSION, oznaczającą, że kursory nie będą rekonstruowane, lub SELECT, która oznacza automatyczną rekonstrukcję kursorów. Zmienna METHOD wskazuje, czy w celu skrócenia czasu awaryjnego przełączenia, każda sesja użytkownika będzie prewencyjnie dublowana, tzn. już przed wystąpieniem awarii zostanie nawiązane połączenie z zapasową instancją. Dostępne wartości zmiennej METHOD to: BASIC, oznaczająca, że nowe połączenie będzie nawiązane dopiero gdy nastąpi awaria, oraz PRECONNECT, oznaczająca, że połączenie zapasowe będzie zawsze otwierane równocześnie z połączeniem głównym. Zmienna RETRIES określa maksymalną liczbę nieudanych prób przełączenia podejmowanych po wystąpieniu awarii. Po przekroczeniu tej liczby aplikacja użytkownika otrzymuje komunikat o błędzie. Zmienna DELAY definiuje odstęp czasowy (w sekundach) pomiędzy kolejnymi próbami nawiązania połączenia. Administrator ma możliwość obserwacji zachowania się mechanizmów TAF za

pomocą perspektywy V\$SESSION – kolumny FAILOVER_METHOD, FAILOVER_TYPE, FAILED_OVER.

6. Strojenie wydajności – problemy typowe

6.1. Rywalizacja o bloki indeksu

Załóżmy, że w bazie danych znajduje się tabela o dużym dziennym przyroście rekordów. Przyjmijmy, że kolumna klucza podstawowego jest wypełniana wartościami całkowitoliczbowymi, wzrastającymi co 1. Na kolumnie klucza podstawowego zdefiniowany jest indeks typu B-drzewo, który jest aktualizowany po każdej operacji wstawienia nowego rekordu. Wstawianie nowych rekordów jest realizowane przez wielu współbieżnych użytkowników, przyłączonych do różnych instancji RAC. W takiej sytuacji należy spodziewać się istotnej degradacji wydajności związanej z koniecznością synchronizowania operacji na blokach indeksu przez wiele instancji RAC. Głównym źródłem tego problemu jest fakt, że sąsiadujące wartości kolumny-klucza są zapisywane w tych samych liściach indeksu.

Typowymi rozwiązaniami problemu rywalizacji o bloki indeksu są: (1) zastosowanie indeksu z odwróconym kluczem (Reverse Key Index), powodującego, że sąsiednie wartości klucza są rozrzucone pomiędzy różne bloki indeksu, (2) zmiana sposobu generowania wartości identyfikatorów tak, aby każda instancja RAC operowała w innym zakresie liczbowym, (3) partycjonowanie indeksu w celu rozproszenia jego danych.

6.2. Generatory sekwencji

Załóżmy, że w bazie danych utworzono generator sekwencji typu NOCACHE ORDER. W środowisku instancji RAC, praca takiego generatora będzie wymagać koordynacji instancji, które pobierają kolejne wartości z generatora. Jeżeli jest to dopuszczalne, to w środowisku RAC należy stosować generatory sekwencji typu CACHE NOORDER.

6.3. Typ przestrzeni tabel

W środowisku RAC zaleca się stosowanie przestrzeni tabel zarządzanych lokalnie, wykorzystujących automatyczne zarządzanie przestrzenią w segmentach.

7. Podsumowanie

W artykule dokonano przeglądu podstawowych elementów architektury Oracle 10g Real Application Clusters oraz omówiono zagadnienia związane z zapewnianiem wysokiej niezawodności pracy.

8. Literatura

1. Dokumentacja techniczna: "Oracle Real Application Clusters Administrator's Guide, 10g Release 1", 2004