

VIII Konferencja PLOUG
Kościelisko
Październik 2002

Technologie klastrowe Oracle

Marcin Przepiórowski

Altkom Akademia S.A.

e-mail: Marcin.Przepiorowski@altkom.com.pl

Abstrakt

Środowiska bazodanowe ulegają szybkiemu rozwojowi. Coraz częściej wykorzystywane są w nich środowiska komputerów równoległych oraz środowiska klastrowe. Firma Oracle wychodząc naprzeciw tym oczekiwaniom, wdraża coraz nowsze rozwiązania równoległego przetwarzania danych. Dokument ten przedstawia systemy równoległe i rozwiązania klastrowe pod kątem serwera Oracle 9i oraz jego poprzednich wersji.

1. Dlaczego klastry?

Wzrastające intensywnie wymagania, dotyczące wielkości baz danych oraz ich wydajności zaczynają odgrywać dużą rolę w planowaniu rozwoju infrastruktury firmy. Jak długo można stale dokupować pamięć RAM lub zmieniać serwery na nowsze?

Działania takie powodują znaczny wzrost kosztów utrzymania środowiska bazodanowego. Korzystniejszym rozwiązaniem jest stworzenie środowiska bazodanowego, które może być skalowane w tańszy i lepszy sposób. Takim rozwiązaniem jest inwestycja w technologie klastrowe, będące w ofercie firmy Oracle od wielu lat.

Rozwiązanie klastrowe zwiększają nie tylko wydajność bazy danych, zwiększają również poziom dostępności systemu bazodanowego. Większa dostępność do danych oznacza mniej niezaplanowanych przestoju w firmie wynikających z awarii serwera bazodanowego, a co się z tym wiąże – mniejsze straty finansowe, które mogłyby wynikać z przestoju bazy danych.

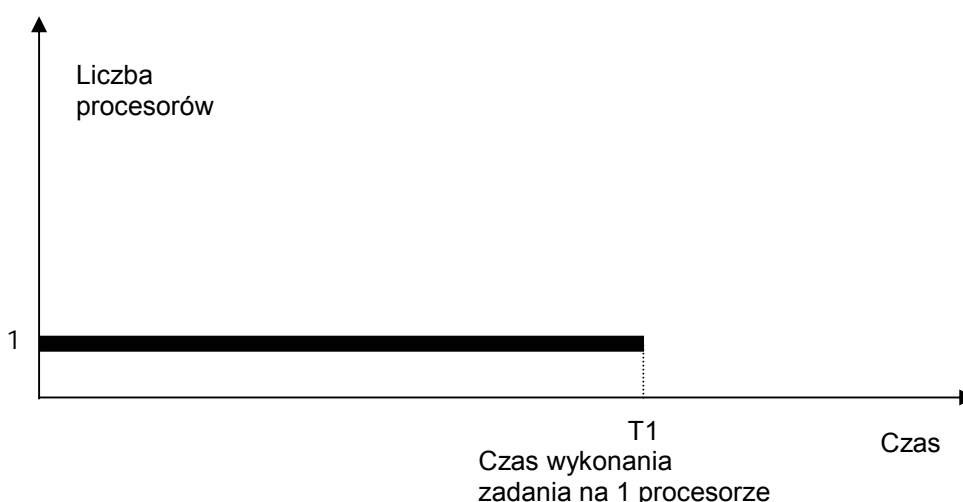
W niniejszym dokumencie opisane zostały mechanizmy przetwarzania równoległego oraz ich zastosowanie w bazach danych na przykładzie serwera Oracle 9i wraz z technologią Real Application Cluster.

Przedstawione w nim zostały najczęściej spotykane architektury komputerów równoległych oraz przetwarzania równoległego w odniesieniu do środowisk, z jakimi spotyka się administrator baz danych. Opisana jest również historia rozwoju przetwarzania równoległego w serwerach Oracle oraz najmlodsze „dziecko” firmy Oracle czyli Real Application Cluster.

2. Komputery równoległe

2.1. Przetwarzanie równoległe

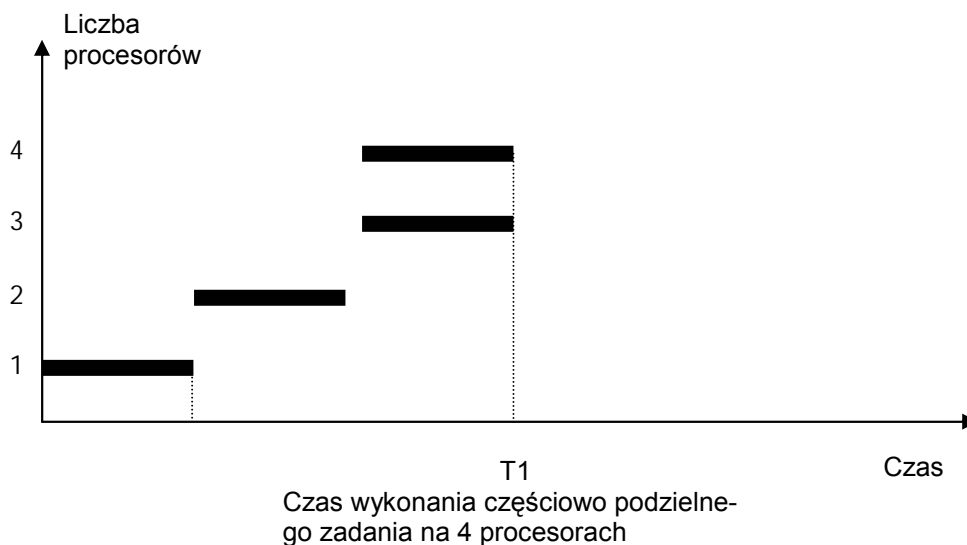
Przetwarzanie równoległe polega po podzieleniu jednego dużego zadania na mniejsze, niezależne od siebie podzadania, a następnie uruchomienie każdego z nich w tym samym czasie na osobnym procesorze (patrz: Rys. 1).





Rys. 1. Rozkładanie obciążenia

Nie każde zadanie można podzielić na mniejsze, niezależne od siebie podzadania, przyrost prędkości wykonania w środowisku do przetwarzania równoległego będzie niewielki (patrz: Rys. 2).



Rys. 2. Nierówne rozłożenie obciążenia

Powyższe przykłady obrazują fakt, że nie wszystkie zadania wykonywane w środowisku równoległym zyskują na czasie realizacji. Jednak oprócz przyspieszenia, systemy równoległe mają wpływ również na skalowalność wykonywanych zadań, można ich wykonać więcej w tym samym czasie (patrz Rys. 3).

Przyspieszenie i skalowalność są terminami zdefiniowanymi matematycznie. Pierwszy z nich określa, ile razy dane zadanie wykonywane jest szybciej w środowisku wieloprocessorowym w stosunku do środowiska jednoprocessorowego.

$$\text{Przyspieszenie} = CJ/CW,$$

gdzie:

CJ – czas wykonania zadania w środowisku jednoprocessorowym,

CW – czas wykonania zadania w środowisku wieloprocessorowym.

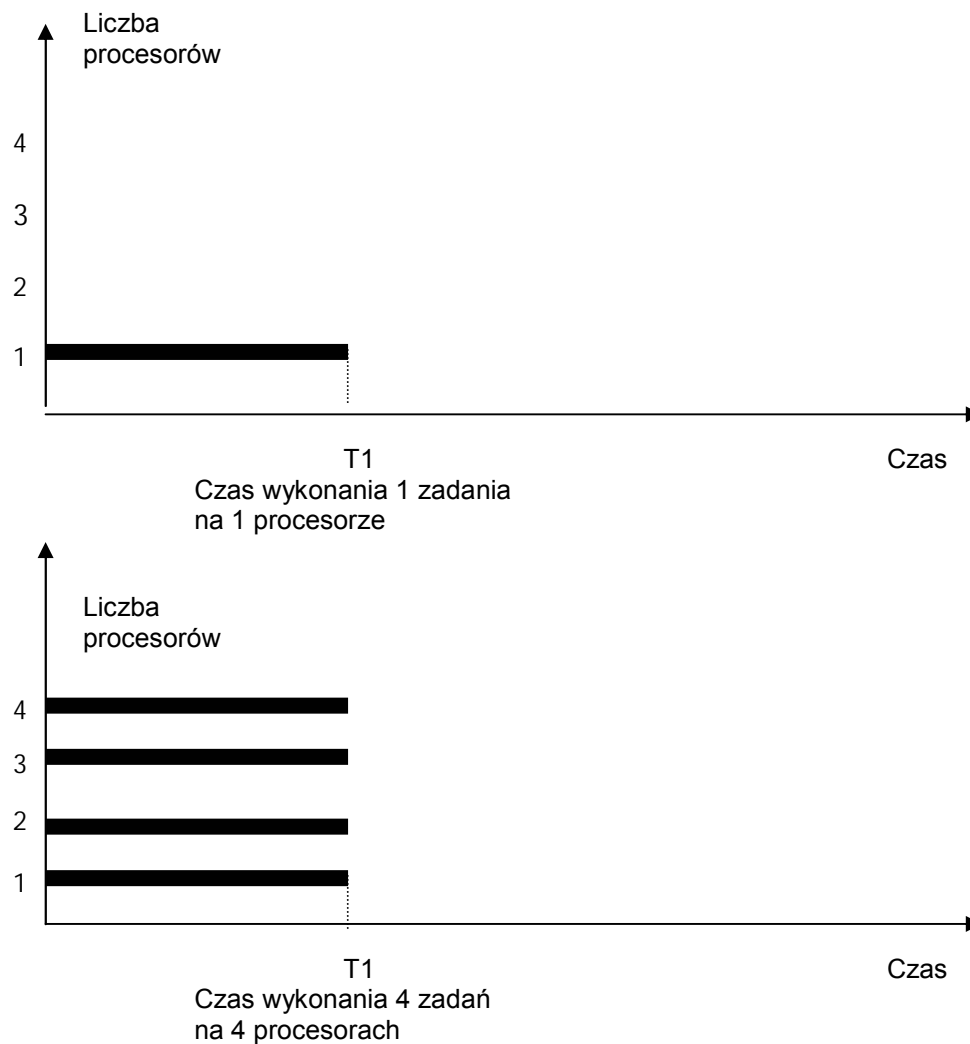
Skalowalność jest zdefiniowana jako stosunek ilości przeprowadzonych transakcji w środowisku wieloprocessorowym do ilości przeprowadzonych transakcji w środowisku jednoprocessorowym w określonej jednostce czasu.

$$\text{Skalowalność} = TW/TJ,$$

gdzie:

TW – ilość transakcji w środowisku wieloprocessorowym,

TJ – ilość transakcji w środowisku jednoprocessorowym.



Rys. 3. Zwiększenie liczby procesorów

2.1.1. Synchronizacja w przetwarzaniu równoległym

Zadania przetwarzane w sposób równoległy wymagają synchronizacji dotyczącej np. dostępu do wspólnych zasobów. Oczekiwanie na komunikację z innym procesem (w celu synchronizacji) obniża wydajność systemów równoległych. W miarę możliwości synchronizacja taka powinna odbywać się za pomocą najszybszego medium transmisyjnego. W pierwszej kolejności za pomocą pamięci operacyjnej, następnie za pomocą połączeń między węzłami, w ostateczności za pomocą pamięci masowej.

Ilość oczekiwań na synchronizację zależy w dużym stopniu od charakteru wykonywanego zadania oraz sposobu implementacji jego w aplikacji.

2.1.2. Zyski z środowiska równoległego w przypadku baz danych

Bazy danych możemy podzielić na dwie podstawowe kategorie:

- OLTP – On-line Transaction Processing,
- DSS – Decision Support System.

Każda z tych kategorii zachowuje się inaczej w środowisku równoległym. Porównanie znajduje się w Tabeli 1.

Tabela 1. Możliwości przyspieszenia i skalowania w bazach danych

Kategoria	Przyspieszenie	Skalowalność
OLTP	Brak	Występuje
DSS	Występuje	Występuje
Mix	Możliwe	Występuje

Zyski, jakie możemy uzyskać z zastosowania środowiska równoległego, to:

- **wyższa wydajność** – zwiększenie liczby procesorów zwiększa skalowalność systemu oraz może zwiększyć szybkość jego działania;
- **wyższe bezpieczeństwo** (dostępność bazy danych) – uszkodzenie jednego węzła w klastrze nie przerywa działania bazy danych;
- **większa elastyczność** – dzięki przezroczystości klastra dla działających aplikacji możemy w dowolnym momencie wyłączać jego węzły;
- **więcej użytkowników** – zwiększenie liczby procesorów umożliwia nam uruchomienie większej ilości procesów serwera a co za tym idzie obsłużenie większej ilości użytkowników.

2.3. Architektura systemów bazodanowych

Środowiska bazodanowe mogą być budowane we wszystkich wymienionych powyżej architekturach sprzętowych oraz w środowiskach klastrowych zbudowanych z połączenia komputerów o dowolnej architekturze. Poniżej przedstawione zostały najpopularniejsze architektury budowy systemów bazodanowych.

Systemy o architekturze „shared memory”

Są to systemy zbudowane o architekturę SMP lub ccNUMA. Baza danych pracująca w tej architekturze jest taka sama jak baza działająca w środowisku jednoprocessorowym. Komunikacja pomiędzy procesorami odbywa się za pośrednictwem pamięci operacyjnej i jest bardzo szybka. Po-

szczególne procesy serwera mogą być obsługiwane przez różne procesory dzięki czemu wzrasta wydajność pracy serwera.

Systemy o architekturze „shared nothing”

Systemy te oparte są o architekturę MPP. Każdy węzeł w tym systemie posiada własną kopię systemu operacyjnego oraz aplikacji. Komunikacja odbywa się za pośrednictwem wewnętrznej sieci. W zależności od konfiguracji bazy danych, może być ona uruchomiona na jednej dużej wirtualnej maszynie lub na każdym węźle uruchomiona jest jedna instancja, które następnie są ze sobą połączone w klastrowy.

Systemy o architekturze „shared disk”

Oparte o dowolną architekturę sprzętową. W systemach tych współdzielony jest dostęp do dysku, na którym znajdują się wspólne dane. Komunikacja pomiędzy węzłami może odbywać się poprzez sieć lub poprzez współdzielony dysk. W systemie istnieje jedna kopia bazy danych obsługiwana przez kilka instancji.

3. Real Application Cluster

Firma Oracle tworząc środowisko bazodanowe działające w środowisku równoległym postawiła na jego ujednoczenie. Baza danych Oracle może działać w środowiskach SMP, MPP oraz ccNUMA. W przypadku uruchomienia jednej instancji Oracle, system operacyjny oraz sprzęt nie mają znaczenia. Z punktu widzenia użytkownika oraz po części administratora RDBMS Oracle wygląda tak samo na wszystkich platformach.

Różnice pojawiają się dopiero w przypadku budowy klastrow. Standardowo klastrow Oracle od wersji 7.3 do 8.1 działa w architekturze „shared disk”, posiadając wszystkie jej wady i zalety. Każdy z węzłów klastra może posiadać dowolną architekturę sprzętowo-programową. Mogą to być serwery wieloprocesorowe oparte o procesory INTEL-a w architekturze SMP, jak również węzeł klastra może być „kawałek” (węzeł) komputera o architekturze MPP. Takie podejście do środowiska umożliwia łatwe zarządzanie i stosunkowo łatwą migrację pomiędzy platformami.

Z kolejnymi wersjami bazy Oracle, zmieniały się mechanizmy wymiany danych w klastrze. Pierwsze klastry komunikowały się głównie przez dysk, obecnie główna wymiana informacji dokonywana jest przez szybką sieć. Klaster bazodanowy w wersji Oracle 9i z opcją Oracle Cache Fusion, wykorzystuje nową architekturę współdzielonych buforów pamięciowych. Dzięki temu rozwiązaniu przełamuje limity tradycyjnych architektur bazodanowych „sharing nothing” i „shared disk”, i umożliwia budowanie wysoce skalowalnych i niezawodnych systemów bazodanowych dla aplikacji e-bussinesu.

Cechy, które powinny kształtować środowisko bazodanowe, zbudowane w oparciu o rozwiązanie klastrowe, to:

- wysoka dostępność – użytkownicy mogą pracować bez względu na awarie sprzętu lub oprogramowania;
- baza danych musi umożliwiać zwiększenie obciążenia, spowodowane rozrostem aplikacji lub ilości danych;
- baza danych musi dobrze obsługiwać różne typy obciążenia, które mogą się zmieniać wraz z zmianami w przedsiębiorstwie;
- serwer musi być dostosowany do rozbudowy.

Wszystkie te cechy posiada Oracle 9i Real Application Cluster, co pozwala na budowanie w oparciu o to środowisko dużych i wydajnych systemów bazodanowych.

Co to jest klaster (wykorzystanie technologii „shared disk”)

Klaster jest grupą niezależnych serwerów, które współpracują ze sobą tworząc jeden wydajny system. Klaster składa się z węzłów klastra (serwerów, ang. *node*), współdzielonych zasobów dyskowych (macierze) oraz połączeń pomiędzy węzłami (interconnect). Węzły w klastrze współdzielą zasoby dyskowe oraz informacje, jednak fizycznie każdy z serwerów (węzłów) posiada swoją odrębną pamięć i system operacyjny oraz aplikacje.

Węzeł może być zbudowany w dowolnej technologii. Może być to zarówno serwer jednoprotocowy, jak i wieloprotocowy (np. w technologii SMP). Na każdym serwerze uruchomiony jest system operacyjny, instancja bazy danych oraz ewentualnie oprogramowanie aplikacyjne.

Połączenie tak zbudowanych węzłów w jeden klaster zwiększa odporność na awarie oraz zwiększa możliwości skalowania bazy danych. Redundancja wszystkich komponentów wchodzących w skład systemu bazodanowego zwiększa jego odporność na awarie.

Sprzęt wykorzystywany w rozwiązaniach klastrowych.

W chwili obecnej sprzęt, który wykorzystywany jest w rozwiązaniach klastrowych podlega ciągłemu rozwojowi. Technologia SAN (*Storage Area Network*) umożliwia dostęp poszczególnym węzłom do dużych ilości dysków. Za pomocą tej technologii, można znieść limity podłączenia ilości dysków do pojedynczego węzła, co pozwala systemom pracującym w technologii „shared disk”, osiągnąć rozmiary danych dostępne do tej pory tylko systemom pracującym w technologii „shared nothing”.

Oprócz rozwoju sprzętu przeznaczonego do przechowywania danych nastąpił również rozwój technologii połączeń i wymiany komunikatów – interconnect. Standard VIA (*Virtual Interface Architecture*) umożliwia budowę szybkich połączeń pomiędzy węzłami, z dobrym współczynnikiem szybkości do ceny.

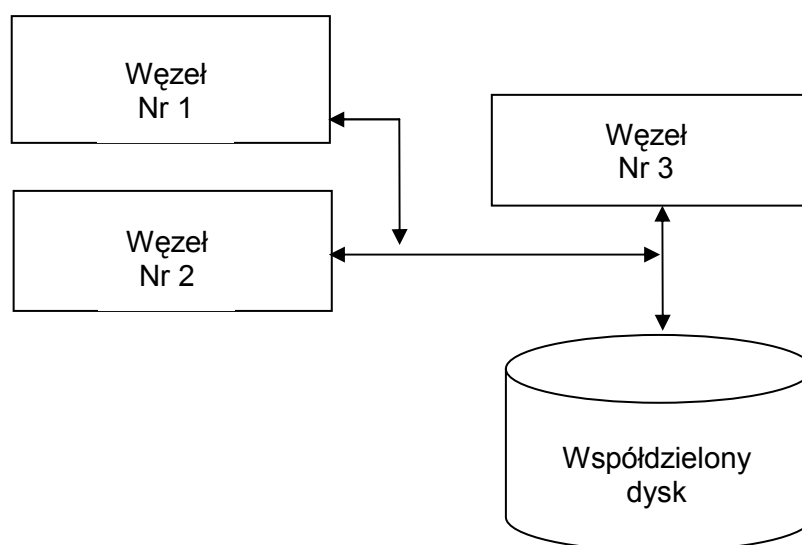
3.1. RAC – architektura

Real Application Cluster (RAC) (patrz: rys. 4) jest środowiskiem sprzętowo-programowym łączącym węzły, na których zostały uruchomione niezależne od siebie instancje serwera Oracle w wersji 9i. Każda instancja może przeprowadzać niezależne transakcje w oparciu o jedną bazę danych. Oprogramowanie RAC zapewnia spójność danych w bazie i zarządza współdzielonym dostępem do danych. Warty podkreślenia jest fakt, że nawet w środowisku klastra cały czas utrzymane jest blokowanie na poziomie wierszy, tak jak ma to miejsce w przypadku jednej instancji.

Real Application Cluster może być stosowany we wszystkich trybach pracy bazy danych. W środowiskach OLTP umożliwia on zwiększenie liczby klientów, którzy mogą być obsługiwani jednocześnie, natomiast w środowiskach DSS zwiększa liczbę przetworzonych wierszy w jednostce czasu.

3.2. Zalety RAC

- **Skalowalność** - umożliwia dołączenie kolejnych węzłów do klastra w celu zwiększenia jego wydajności. Ilość węzłów, które można dołączyć zależy od platformy, na jakiej zainstalowany jest serwer.
- **Wysoka niezawodność** - zapewnienie niezawodności działania środowiska w przypadku problemów natury sprzętowej lub programowej jednego z elementów klastra. Poszczególne węzły są od siebie odizolowane, więc awaria jednego z nich nie wpływa na pracę drugiego.
- **Przezroczystość** - z punktu widzenia aplikacji klaster wygląda i zachowuje się tak samo jak pojedyncza instancja.



Rys. 4. Architektura Real Application Cluster

3.3. Porównanie RAC i Parallel Servera

W wersjach poprzednich systemu Oracle (7.x.x oraz 8.x.x) połączenie kilku instancji nazywane było Serwerem Równoległym (*Parallel Server*). Jego implementacja często wiązała się z koniecznością dostosowania aplikacji. Przyczyną była wymiana informacji synchronizujących przez dysk, co miało zły wpływ na wydajność. Mechanizm "cache fusion" – usuwający tę wadę – pojawił się w po raz pierwszy w wersji 8.1.x, ale nie był jeszcze w pełni funkcjonalny. Obsługiwał on tylko żądania typu read/read lub read/write, natomiast żądania typu write/write dalej były obsługiwane poprzez dyskowy kanał I/O.

Aby uniknąć wymiany danych przez dyskowy kanał I/O, projektanci aplikacji musieli dokonywać jej partycjonowania, co często wiązało się z dodatkowym kosztem wdrożenia lub utrzymania aplikacji. Pojawienie się wersji 9i wraz z pełną implementacją mechanizmu "cache fusion", umożliwiło wydajną pracę każdej aplikacji w środowisku klastrowym bez konieczności jej przebudowy. Wszystkie żądania dostępu do bloku, który znajduje się w pamięci cache dowolnej instancji obsługiwane są teraz bez potrzeby komunikacji poprzez dyskowy kanał I/O. Wynika z tego, że liczba zapisów na dysk w wersji 9i jest taka sama w przypadku działania bazy w architekturze klastrowej jak i w architekturze pojedynczej instancji. Dzięki temu skalowanie klastra nie wpływa na prędkość działania aplikacji.

Tabela 2. Porównanie Real Application Cluster i Parallel Server

Real Application Cluster	Oracle Parallel Server
skalowalność, bez wpływu na wydajność	Skalowalność, ograniczona koniecznością wymiany informacji poprzez kanał dyskowy
zapewnienie wysokiej dostępności	zapewnienie wysokiej dostępności
zwiększenie szybkości działania dla wszystkich aplikacji	zwiększenie szybkości działania dla aplikacji partycjonowanych

3.4. Architektura Real Application Clustra

W skład klastra wchodzi następujące elementy:

- Cluster Monitor;
- Global Cache Service oraz Global Enqueue Service;
- Cluster Interconnect oraz komunikacja międzyprocesowa;
- podsystemy dyskowe.

3.4.1. Cluster Monitor

Monitor klastra nadzoruje pracę całego klastra. W każdej chwili posiada on aktualny obraz klastra, tzn. informacje o stanie wszystkich nadzorowanych węzłów oraz instancji uruchomionych na tych węzłach. Jego podstawową rolą jest zapewnienie wysokiej dostępności klastra oraz ochrona danych przed uszkodzeniem. W przypadku, gdy którakolwiek z instancji zacznie działać w sposób niekontrolowany, zostanie ona odcięta od bazy danych i zatrzymana. Przyczyny odcięcia instancji od bazy danych oraz jej zatrzymania:

- zatrzymanie jednego z procesów instancji;
- awaria węzła;
- odłączenie się węzła od reszty klastra.

W sytuacji awaryjnego zatrzymania działania instancji, proces odtwarzania danych zawartych w dziennikach powtórzeń tej instancji jest wykonywany automatycznie i w sposób niewidoczny dla użytkowników. Jednocześnie następuje rekonfiguracja klastra i odłączenie nieczynnej instancji z zasobów klastra. O fakcie tym zostaje poinformowany podsystem Global Cache Service, który od tego momentu przestaje korzystać z zasobów nieaktywnej instancji.

Częścią Cluster Monitora odpowiedzialną za nadzorowanie zasobów sprzętowych węzłów oraz komunikację pomiędzy węzłami jest Node Monitor – przeważnie oprogramowanie klastrowe dostarczane przez producenta sprzętu lub oprogramowania. Są to procesy silnie powiązane z warstwą systemu operacyjnego z racji konieczności dostępu do wielu informacji systemowych. Node Monitor nadzoruje pracę wszystkich podsystemów węzła ważnych dla działania na nim instancji Oracle. Kontroluje on również dostępność w danej chwili węzłów w klastrze, przekazując informacje o nich do innych podsystemów Cluster Monitora.

3.4.2. Podsystemy Global Cache Service i Global Enqueue Service

Global Cache Service oraz Global Enqueue Service są integralnymi częściami technologii Real Application Cluster. Zapewniają koordynację pomiędzy poszczególnymi instancjami w czasie dostępu do współdzielonych zasobów. Celem ich jest zapewnienie spójności danych w bazie, w przypadku gdy więcej niż jedna instancja korzysta ze współdzielonego zasobu.

Podstawowe cechy obu podsystemów:

- **przezroczystość** – aplikacja kliencka nie „widzi” działania tych systemów; w celu zapewnienia spójności danych aplikacja powinna korzystać z tych samych mechanizmów, jak w przypadku działania w środowisku jednej instancji;
- **rozproszona architektura** – oba podsystemy przechowują informacje w Global Resource Directory, który jest umieszczony w pamięci każdej instancji, a wszelkie zmiany dokonane przez którąkolwiek z instancji są natychmiast dystrybuowane do innych kopii tego katalogu;
- **odporność na błędy** – awaria jednej instancji, a co z tym idzie jednej kopii katalogu Global Resource Directory nie powoduje żadnych skutków ubocznych, ponieważ inne instancje

nadal posiadają kopie tego katalogu. Dzięki temu do momentu istnienia dostępu do przynajmniej jednej instancji, przechowywany jest cały katalog;

- **zarządzanie zasobami** – oba podsystemy używają jednej instancji do zarządzania jednym konkretnym zasobem. Przypisanie zasobu do instancji, która najczęściej z niego korzysta odbywa się w celu optymalizacji tego procesu. W przypadku, gdy inna instancja zaczyna częściej korzystać z danego zasobu, zarządzanie tym zasobem przenosi się zawsze do tej instancji, która statystycznie częściej wymaga do niego dostępu.

3.4.3. Cluster Interconnect oraz komunikacja międzyprocesowa

Real Application Cluster korzysta z komunikacji międzyprocesowej w oparciu o protokół IPC. Za pomocą tego protokołu następuje komunikacja pomiędzy składnikami klastra w ramach jednego węzła, jak również pomiędzy węzłami. Każde połączenie składa się z niezależnych wiadomości, które są przesyłane asynchronicznie i następnie kolejgowane w węzłach. Komunikacja IPC jest zaprojektowana z intencją przesyłania komunikatów tak szybko jak pozwoli na to sieć, za pomocą której połączone są węzły.

3.4.4. Podsystemy dyskowe.

Wspólne podsystemy dyskowe muszą być dostępne z każdego węzła, ponieważ na nich znajduje się współdzielona baza danych. Serwer Oracle może korzystać z urządzeń surowych (*raw devices*) lub klastrowego systemu plików.

3.5. „Cache Fusion” i podsystem Global Cache Service

Mechanizm „cache fusion” jest nową technologią wymiany informacji pomiędzy buforami instancji Oracle uruchomionych w klastrze. Do komunikacji używany jest protokół IPC oraz szybkie łącza pomiędzy węzłami klastra. Mechanizm ten pozwala wyeliminować zbędne operacje dyskowe, które ze względu na czas ich wykonania mają duży wpływ na wydajność systemu. Blok bazodanowy, który został odczytany przez jedną z instancji i znajduje się w jej buforze, może być odczytywany przez inne instancje klastra bezpośrednio z buforów tej instancji. Mechanizm „cache fusion” w wersji 9i obsługuje wszystkie rodzaje jednoczesnego dostępu do bloku:

- read/read – dwie instancje żądają danego bloku do odczytu;
- read/write – jedna instancje żąda bloku do modyfikacji, druga żąda bloku do odczytu;
- write/write – obie instancje żądają bloku do modyfikacji.

Poniżej przedstawiono schematy działania opisujące różne tryby dostępu:

- **Jednoczesny odczyt przez różne instancje – *read/read***
Dostęp w trybie read/read jest najprostszy do rozwiązania przez systemy Real Application Cluster-a. Wiele instancji może współdzielić jeden blok, w swoich buforach i jeżeli nie dokonują jego modyfikacji mogą mieć zawsze dostęp do tego bloku, bez potrzeby synchronizowania buforów w klastrze.
- **Jednoczesny odczyt i zapis przez różne instancje – *read/write***
Dostęp w trybie write/read dość często występuje w aplikacjach typu OLTP. Jedna z instancji (A) zmodyfikowała zawartość bloku bazodanowego przechowywanego w swoim buforze, w tym samym czasie inna instancja (B) chce odczytać dany blok bazodanowy. Instancja B otrzyma spójną pod względem odczytu kopię bloku bazodanowego bezpośrednio z bufora instancji A. Za zapewnienie spójności odczytu odpowiedzialny jest proces podsystemu Global Cache Service – LSMn. Spójność odczytu, przeprowadzana jest za pomocą obrazów bloków (Past Image).

- **Jednoczesny zapis przez różne instancje – *write/write***
Dostęp do współdzielonego bloku bazodanowego w trybie *write/write* bez użycia to tego celu operacji dyskowych został wprowadzony w wersji 9i Oracle-a. Wymiana bloków pomiędzy instancjami odbywa się tylko i wyłącznie poprzez protokół IPC i połączenia między węzłowe. Jeżeli instancja A zmodyfikowała blok, który następnie ma być zmodyfikowany przez instancję B, to przed przesłaniem takiego bloku z instancji A do B, w instancji która zmodyfikowała blok zapamiętywana jest aktualna wersja bloku (Past Image), i jednocześnie następuje zmiana trybu współdzielenia bloku z trybu lokalnego na globalny. Blok przed ani po przesłaniu nie jest zapisywany na dysk, co znacznie przyspiesza działania związane z modyfikacją bloków. Proces DBWR zapisuje bloki na dysk, tak samo jak w przypadku pracy jednej instancji.

Obraz bloku bazodanowego, który został zmodyfikowany przez instancję nazywany jest „Past Image”. Tworzony on jest przed przesłaniem bloku bazodanowego do innej instancji, która chce dokonać jego modyfikacji. Past Image jest przechowywany przez każdą instancję, która dokonała zmian i jest zapisywany w jej redo logach. Ma to na celu przyspieszenie odtworzenia bazy, w przypadku awarii instancji do której przesłano blok, a w związku z tym z koniecznością wycofania nie zatwierdzonych transakcji. Kolejnym zadaniem mechanizmu „Past Image” jest prostsze i szybsze zapewnienie spójności odczytu.

3.6. Load Balancing i Transparent Application Failover

Rozwiązanie oparte o Real Application Cluster pozwala na wdrożenie dwóch mechanizmów usprawniających pracę użytkownika. Jest to Load Balancing pozwalający na równomierne obciążenie pracą wszystkich serwerów pracujących w klastrze oraz Transparent Application Failover (TAF) pozwalający na przenoszenie sesji użytkownika z serwera, który uległ awarii na inny serwer pracujący w klastrze. Mechanizmy te wymagają odpowiedniej konfiguracji warstwy sieciowej Oracle i są przez nią realizowane. Konfiguracja Net8 polega (w przypadku rozkładania obciążenia) tylko i wyłącznie na konfiguracji stacji klienckiej. W przypadku TAF należy właściwie skonfigurować proces nasłuchowy, instancję oraz stację kliencką.

Mechanizm „Load Balancing” działa na zasadzie sprawdzania aktualnego obciążenia instancji i przyjmowania lub przekierowania do innej instancji kolejnego nadchodzącego połączenia. Dzięki temu wszystkie instancje w klastrze są równo obciążone.

Mechanizm „TAF” działa w momencie podłączania się do serwera oraz w czasie sesji użytkownika. Jeżeli zostanie wykryta awaria, to następuje automatyczne przeniesienie sesji do innej instancji. W wyniku przeniesienia sesji aktualnie trwająca transakcja jest wycofywana i zerowane są zmienne PL/SQL-a. Jednocześnie aplikacja dostaje za pomocą biblioteki OCI informacje o przeniesieniu sesji. Tą samą informację można odczytać z perspektywy systemowej *V\$SESSION*. Aplikacja, która działa w środowisku klastrowym powinna wykrywać fakt przeniesienia sesji i odpowiednio na ten fakt zareagować, np. ponowieniem wycofanej transakcji. Takie rozwiązanie, gwarantuje pełną przejrzystość dla użytkownika aplikacji, który nie jest świadomy faktu awarii serwera i przeniesienia jego sesji na inny serwer.

Bibliografia

1. Oracle Concepts 9.0.1, part A89867-01
2. Instalation and configuration 9.0.1, part A89868-01
3. Administration 9.0.1, part A89869-01
4. Oracle Metalink, note 139436