

X Konferencja PLOUG  
Kościelisko  
Październik 2004

# Serwer aplikacji OracleAS 10g w architekturach o podwyższonej niezawodności

Maciej Zakrzewicz

*PLOUG, Politechnika Poznańska  
mzakrz@cs.put.poznan.pl*

## **Streszczenie**

Jednym z etapów wdrażania wielowarstwowych systemów informatycznych jest przygotowanie odpowiednio skonfigurowanej platformy serwera aplikacji. Obecne zastosowania coraz częściej stawiają znaczne wymagania w zakresie dostępności i niezawodności serwerów aplikacji. Wysoka niezawodność serwera aplikacji może być osiągnięta poprzez użycie właściwie dobranej architektury sprzętowo-programowej. W artykule omówiono metody podnoszenia niezawodności Oracle Application Server 10g m.in. za pomocą klastrów serwerów aplikacji (zarządzanych automatycznie z użyciem różnych typów repozytoriów i zarządzanych ręcznie) oraz klastrów infrastruktury (Cold Failover Cluster, Active Failover Cluster, Identity Management Service Replication).



## 1. Wprowadzenie

Serwer aplikacji to oprogramowanie o charakterze systemowym, stanowiące platformę wykonywania komponentów aplikacji wielowarstwowych. Serwer aplikacji oferuje wykonywanym programom szereg usług m.in. obsługi komunikacji z komponentami klienta, komunikacji z bazami danych, uwierzytelniania i autoryzacji użytkowników, rozpraszania żądań i obciążenia, itp. Łączność komponentów klienta z serwerem aplikacji jest zwykle realizowana z użyciem protokołu HTTP lub HTTPS (HTTP szyfrowany przez SSL). Na rynku dostępnych jest wiele amatorskich i komercyjnych serwerów aplikacji – do czołówki komercyjnej niewątpliwie należą: BEA WebLogic, IBM WebSphere i Oracle Application Server.

Niezawodność działania serwera aplikacji jest wyznacznikiem niezawodności całego systemu informatycznego, pracującego na jego platformie. Z chwilą, gdy awarii ulega serwer aplikacji, klienci tracą dostęp do swoich aplikacji biznesowych, w wyniku czego funkcjonowanie firmy lub instytucji ulega całkowitemu paraliżowi. Obecne zastosowania zatem coraz częściej stawiają znaczne wymagania w zakresie dostępności i niezawodności serwerów aplikacji.

Z punktu widzenia niezawodności serwerów aplikacji należy wyodrębnić trzy główne źródła awarii: (1) uszkodzenia pojedynczych procesów serwera aplikacji wynikające z błędów oprogramowania, (2) uszkodzenia sprzętowe, np. pamięci operacyjnej lub CPU, powodujące awarię systemu operacyjnego, (3) ataki zewnętrzne typu DOS (Denial of Service), których celem jest zablokowanie funkcjonowania serwera aplikacji poprzez skierowanie do niego nadmiernego obciążenia. Zwykle najtrudniejsze do rozwiązania są awarie typu (2), gdyż w istocie wymagają stosowania sprzętu „bezawaryjnego”. Ze względu na bardzo wysoki koszt sprzętu komputerowego, który posiada wewnętrzne zabezpieczenia przed skutkami fizycznych uszkodzeń, dużą popularnością cieszą się architektury oparte na redundancji sprzętowej, nazywane powszechnie architekturami klastrowymi. W architekturach takich elementy serwera aplikacji są rozproszone pomiędzy wiele fizycznych maszyn w taki sposób, aby po awarii jednej z nich, pozostałe mogły przejąć obsługę wszystkich żądań użytkowników.

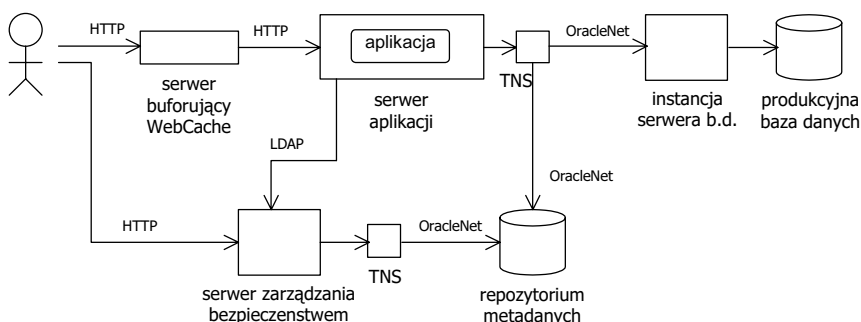
W dalszej części artykułu omówimy techniczne rozwiązania serwera aplikacji Oracle Application Server 10g (OAS 10g) umożliwiające konstrukcję klastrów podnoszących niezawodność systemu.

## 2. Podstawowa architektura OAS 10g

Oracle Application Server 10g jest produktem, na który składają się cztery główne elementy: serwer aplikacji (Middle Tier), serwer zarządzania bezpieczeństwem (Identity Management), repozytorium metadanych (Metadata Repository) oraz serwer buforujący WebCache. Zasadniczym składnikiem jest serwer aplikacji, będący środowiskiem wykonywania kodu programów wielowarstwowych, takich jak aplikacje CGI, FastCGI, J2EE, Oracle Forms, Oracle Reports, Oracle Discoverer, Oracle Portal. Serwer aplikacji ma budowę modułową i zawiera m.in. serwer HTTP (OHS, Oracle HTTP Server) i serwer J2EE (OC4J, Oracle Containers for J2EE). Serwer zarządzania bezpieczeństwem służy do centralnego uwierzytelniania użytkowników oraz obsługi ich certyfikatów klucza publicznego. Repozytorium metadanych jest bazą danych gromadzącą zarówno parametry konfiguracyjne serwerów aplikacji, parametry związane z bezpieczeństwem (hasła, uprawnienia), jak i kod programowy aplikacji realizowanych w niektórych technologiach, np. Oracle Discoverer lub Oracle Portal. Często serwer zarządzania bezpieczeństwem wraz z repozytorium metadanych są wspólnie nazywane infrastrukturą (Infrastructure). Serwer buforujący WebCache odpowiada za buforowanie statycznych i dynamicznych dokumentów WWW, dzięki czemu wiele żądań użytkowników może być obsługiwanych bez konieczności wykonywania kodu na serwerze aplikacji. Należy podkreślić, że w wielu konfiguracjach nie jest wymagane posiadanie

serwera zarządzania bezpieczeństwem ani repozytorium metadanych, ani serwera buforującego WebCache.

Przykładowy przebieg obsługi żądania użytkownika został przedstawiony na rys. 1. Użytkownik wysyła żądanie HTTP do serwera buforującego WebCache. Jeżeli WebCache nie posiada w pamięci gotowego dokumentu stanowiącego odpowiedź na takie żądanie, to żądanie jest przekazywane do serwera aplikacji. Z chwilą otrzymania żądania, serwer aplikacji dokona weryfikacji użytkownika oraz sprawdzi jego uprawnienia, posługując się serwerem zarządzania bezpieczeństwem. Krok ten może wymagać przekierowania użytkownika do serwera zarządzania bezpieczeństwem w celu pobrania nazwy i hasła użytkownika. Serwer zarządzania bezpieczeństwem orzeka o poprawności hasła użytkownika w oparciu o zapisy w repozytorium metadanych. Po pomyślnym uwierzytelnieniu użytkownika serwer aplikacji uruchamia kod żądanego programu. Gdyby kod programu znajdował się w repozytorium metadanych (kod PL/SQL), wtedy serwer aplikacji otworzyłby bezpośrednie połączenie z repozytorium i wywołał żadaną jednostkę programową. Jeżeli podczas wykonywania kodu na serwerze aplikacji zaistnieje potrzeba komunikacji z produkcyjną bazą danych, to stosowne połączenie zostanie otwarte przez serwer aplikacji z użyciem adaptera Oracle Net.



Rys. 1. Obsługa żądania użytkownika przez Oracle Application Server 10g

### 3. Mechanizmy zapewniania niezawodności w instalacjach jednomaszynowych

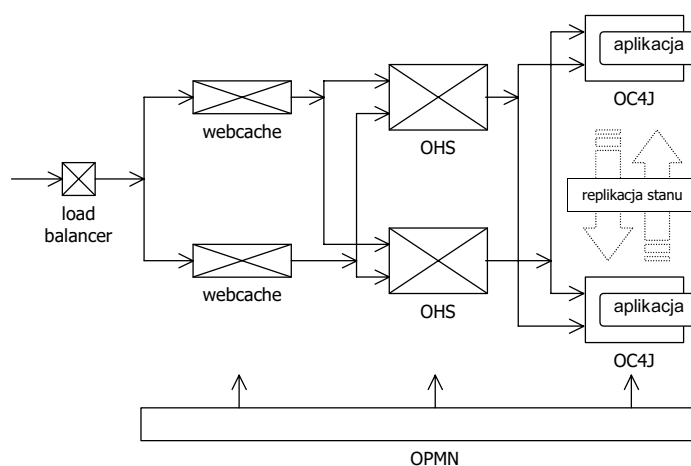
Gdy cała instalacja OAS 10g znajduje się na pojedynczej maszynie, wtedy z oczywistych powodów system taki jest podatny na awarie o charakterze sprzętowym. Uszkodzenia CPU, pamięci operacyjnej, sterowników dyskowych, itd. uniemożliwiają dalsze funkcjonowanie systemu i powodują zatrzymanie obsługi żądań użytkowników. Niemniej jednak, nawet dla takiej instalacji administrator może skorzystać z mechanizmów zapewniania niezawodności, chroniących system przed skutkami awarii o charakterze programowym, jak np. awaria pojedynczego procesu serwera. Poniżej przedstawimy mechanizmy podnoszenia niezawodności i samonaprawiania komponentów serwera aplikacji (rys. 2).

W celu poprawy dostępności serwera aplikacji w przypadku, gdy awarii ulega jeden z jego procesów, OAS 10g oferuje mechanizm zwielokrotniania procesów. Zwielokrotnienie procesów może dotyczyć zarówno serwera HTTP, serwera J2EE, jak i wielu innych komponentów odpowiedzialnych za wykonywanie aplikacji. Zwielokrotnienie procesów oznacza, że w normalnych warunkach pracy, żądania użytkowników są rozpraszane pomiędzy wiele alternatywnych procesów wykonawczych. Gdy jeden z nich ulegnie awarii, wtedy pozostałe mogą kontynuować obsługę użytkowników.

Zarówno serwer aplikacji, jak i serwer zarządzania bezpieczeństwem, zawierają komponent służący do ciągłej obserwacji aktywności wszystkich procesów oraz podejmowania akcji ratunkowych: OPMN (Oracle Process Management and Notification). Na żądanie administratora OPMN uruchamia procesy serwera aplikacji lub serwera zarządzania bezpieczeństwem, a następnie moni-

toruje ich pracę w celu gromadzenia danych o wydajności oraz wykrywania awarii. Jeżeli OPMN wykryje awarię procesu, wtedy informuje o niej inne procesy zależne i przystępuje do próby ponownego uruchomienia procesu. Dzięki temu, procesy serwera aplikacji i serwera zarządzania bezpieczeństwem są „nieśmiertelne”, gdyż po każdej ich awarii następuje (w miarę możliwości) samonaprawa.

Samonaprawianie komponentów serwera aplikacji nie zawsze pozwala na nieprzerwaną pracę użytkowników. Jeżeli awarii ulegnie proces wykonujący aplikację (np. OC4J), wtedy utracie ulega jej stan (wartości zmiennych, obiekty), nawet jeżeli proces zostanie uruchomiony ponownie. W celu ochrony przed utratą stanu aplikacji OAS 10g umożliwia stosowanie mechanizmu replikacji stanu aplikacji (state replication), polegającego na powielaniu stanu każdej wykonywanej aplikacji pomiędzy wiele procesów wykonujących aplikację. Obecnie mechanizm ten dostępny jest wyłącznie dla procesów serwera J2EE.



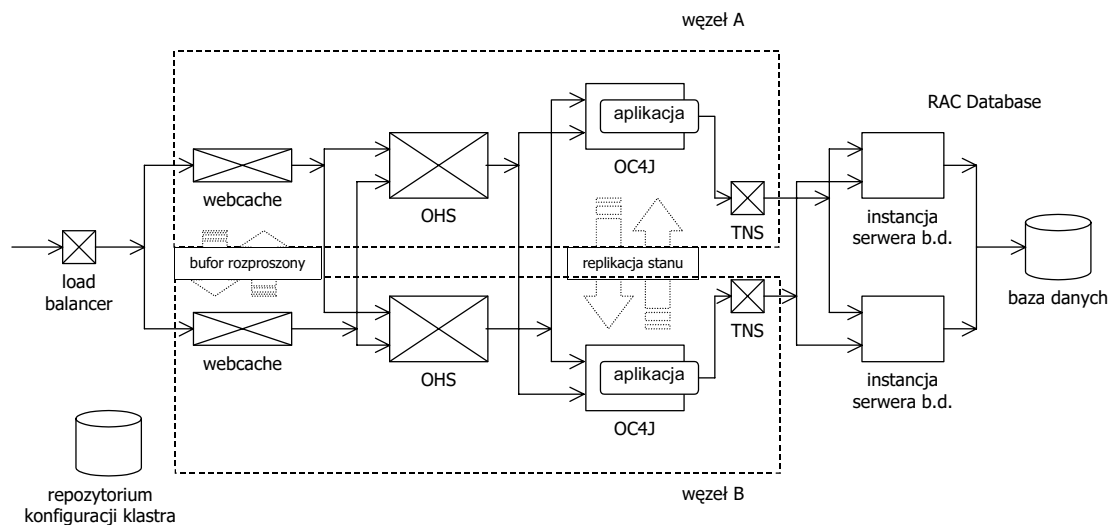
Rys. 2. Jednomaszynowa instalacja OAS 10g o podwyższonej niezawodności

## 4. Mechanizmy zapewniania niezawodności w instalacjach wielomaszynowych

### 4.1. Klastery serwerów aplikacji

OAS 10g umożliwia konstrukcję klastrów serwerów aplikacji, w których obsługa żądań użytkowników jest rozproszona pomiędzy wiele równoważnych maszyn. Każdy z serwerów uczestniczących w klastrze musi posiadać identyczną konfigurację aplikacji, gdyż musi potrafić zastąpić dowolny inny serwer w tym samym klastrze. W celu ułatwienia administracji klastrami serwerów aplikacji, OAS 10g pozwala scentralizować konfigurację wszystkich serwerów aplikacji poprzez jej zapis we wspólnym repozytorium. Repozytorium konfiguracji klastra może znajdować się w repozytorium metadanych (Metadata Repository) lub w systemie plików wybranego serwera wchodzącego w skład klastra (File-based Repository). Narzędzia administracyjne OAS 10g umożliwiają m.in. automatyczną instalację aplikacji na wszystkich maszynach klastra, zdalną rekonfigurację dowolnej maszyny wchodzącej w skład klastra.

Dla sprawnego funkcjonowania klastrów serwerów aplikacji niezbędna jest obecność mechanizmów rozpraszania żądań użytkowników oraz mechanizmów replikacji stanu aplikacji (rys. 3). Poniżej przedstawimy dostępne rozwiązania w zakresie rozpraszania żądań.



Rys. 3. Architektura klastra serwerów aplikacji

## 4.2. Mechanizmy rozpraszania żądań (load balancing)

### 4.2.1. WebCache

Moduł WebCache umożliwia przekazywanie żądań HTTP do wielu serwerów aplikacji. W konfiguracji WebCache z każdym serwerem aplikacji związana jest liczba całkowita (capacity) określająca maksymalną liczbę równoczesnych żądań, jakie serwer aplikacji może przyjąć od WebCache. Liczba ta wpływa również na względny rozkład żądań pomiędzy serwerami aplikacji – przykładowo, jeżeli dla serwera aplikacji A wartość Capacity wynosi 100, a dla serwera aplikacji B wynosi 200, to serwer B będzie otrzymywać 66,6 % żądań a serwer A tylko 33% żądań użytkowników. WebCache wstrzyma przekazywanie żądań do serwera aplikacji, jeżeli stwierdzi jego awarię. Nieaktywny serwer aplikacji będzie wówczas okresowo odpytywany przez WebCache, aby wykryć moment jego ponownego włączenia.

### 4.2.2. Serwer HTTP

Gdy serwer HTTP otrzymuje żądania wykonania aplikacji J2EE, wtedy przy użyciu protokołu AJP przekazuje je do jednego z pracujących serwerów J2EE (OC4J). Za bezpośrednią komunikację odpowiada wewnętrzny moduł serwera HTTP, nazywany mod\_oc4j. W celu uniknięcia kierowania żądań do niesprawnych lub niepracujących serwerów J2EE, każdy serwer HTTP posiada tablicę routingu (routing table), której wpisy reprezentują dostępne w danej chwili serwery J2EE. Za aktualizację tablicy routingu odpowiada proces OPMN, który w przypadku wykrycia awarii procesu serwera J2EE usunie go z tej tablicy. Podobnie, po pomyślnym uruchomieniu procesu serwera J2EE, OPMN wprowadzi go do tablicy routingu serwera HTTP.

Wybór procesu serwera J2EE, do którego zostanie przekazane żądanie jest dokonywany za pomocą jednego z kilku algorytmów, wybranego przez administratora: Round Robin, Random lub Metric-Based. W algorytmie Round Robin, kolejne żądania są w stałym porządku przekazywane do kolejnych procesów serwera J2EE. Algorytm Random każdorazowo dokonuje losowego wyboru procesu serwera J2EE, do którego zostanie przekazane żądanie użytkownika. Natomiast algorytm Metric-Based wykorzystuje informacje o aktualnym obciążeniu każdego z procesów serwera J2EE i dokonuje wyboru tego z nich, który jest obciążony najmniej.

Algorytmy rozpraszania żądań mogą również korzystać z predefiniowanych wag węzłów (weighted routing) oraz preferować przetwarzanie lokalne (local affinity). Predefiniowanie wag węzłów polega na określeniu przez administratora mocy obliczeniowej każdego węzła jako pewnej liczby całkowitej, która jest wykorzystywana do wyboru procesu serwera J2EE w taki sposób, aby

węzły silniejsze obliczeniowo otrzymywały więcej żądań użytkowników. Natomiast preferowanie przetwarzania lokalnego polega na skłonności serwera HTTP do wyboru procesów serwera J2EE znajdujących się na tym samym węźle.

#### **4.2.3. Oracle Net**

Biblioteka komunikacji sieciowej Oracle Net dopuszcza konfiguracje, w których z pojedynczą nazwą usługi związane kilka instancji bazy danych. W takim przypadku, każdorazowo podczas nawiązywania połączenia z bazą danych następuje wylosowanie instancji, do której zostanie skierowane żądanie. Gdyby instancja ta była niedostępna, wtedy losowanie zostanie powtórzone i nastąpi kolejna próba nawiązania połączenia. Tego typu konfiguracja Oracle Net jest powszechnie wykorzystywana w architekturach serwera bazy danych RAC (Real Application Clusters), gdzie wszystkie instancje są równoważne i oferują jednakowy dostęp do bazy danych.

### **4.3. Klastry WebCache**

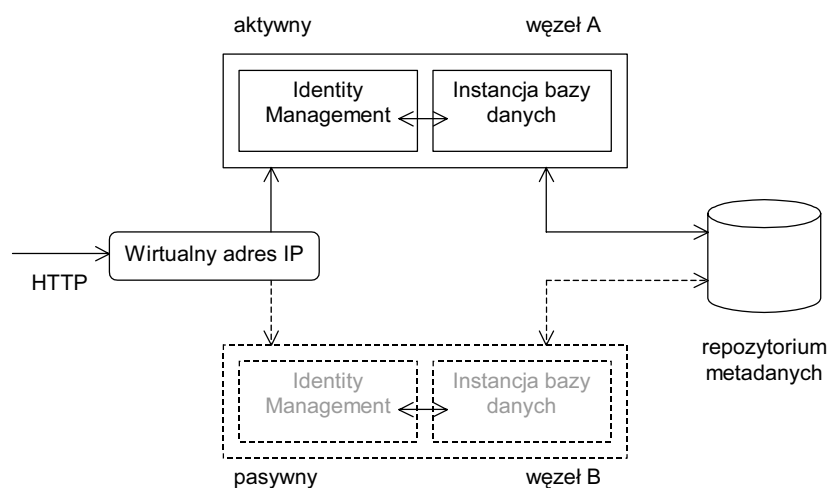
Serwery buforujące WebCache mogą być organizowane w klastry, w ramach których ich pamięć buforowa jest rozproszona pomiędzy wiele maszyn. W takiej konfiguracji każdy z serwerów buforowych odpowiada za dostarczanie tylko pewnego podzbioru dokumentów, wyznaczonego przez wartość funkcji haszowej opartej na adresie URL. Jeżeli żądanie użytkownika trafia do serwera buforującego, który nie odpowiada za obsługę tego żądania, wtedy serwer ten przekaże żądanie właściwemu serwerowi. W przypadku awarii jednego serwera buforującego w klastrze, pozostałe serwery reorganizują metodę podziału dokumentów.

### **4.4. Klastry serwerów zarządzania bezpieczeństwem**

#### **4.4.1. Cold Failover Cluster**

Architektura ta, nazywana również układem „aktywny-pasywny”, składa się z dwóch węzłów korzystających ze wspólnej, współdzielonej pamięci dyskowej (rys. 4). Węzły odpowiadają za obsługę komponentów IM oraz za obsługę instancji bazy danych repozytorium metadanych. W danej chwili tylko jeden z węzłów jest aktywny – drugi pozostaje wyłączony. W przypadku awarii węzła aktywnego, należy uruchomić węzeł dodatkowy i za jego pomocą kontynuować obsługę żądań użytkowników. Przełączenie węzłów może być wykonane manualnie przez administratora lub z użyciem zautomatyzowanego oprogramowania, dostarczanego przez dostawcę sprzętu.

Konfiguracja Cold Failover Cluster wymaga korzystania z wirtualnego adresu IP i wirtualnej nazwy węzła, które są dynamicznie odwzorowywane na adresy fizyczne. Dzięki temu, przełączenie węzła aktywnego w przypadku awarii nie wymaga żadnych zmian konfiguracji serwerów aplikacji, które korzystają z infrastruktury. Węzły muszą posiadać identyczną instalację i konfigurację oprogramowania.

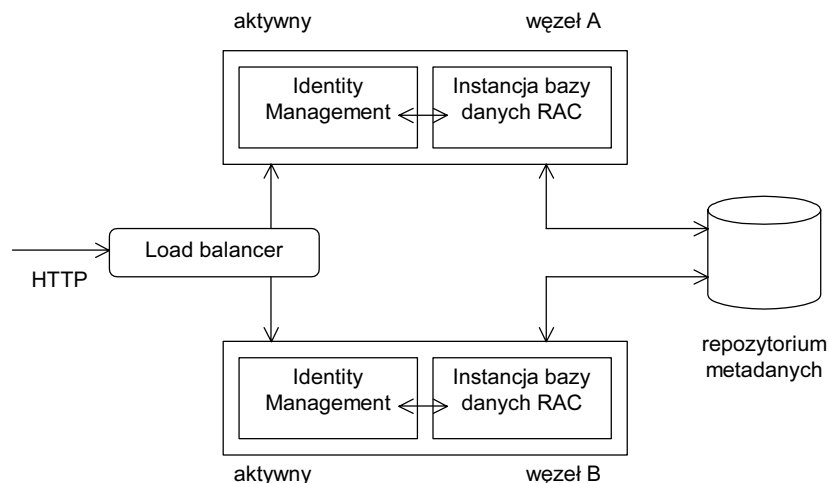


Rys. 4. Infrastruktura w architekturze Cold Failover Cluster

#### 4.4.2. Active Failover Cluster

Architektura ta, nazywana również układem „aktywny-aktywny”, jest podobna do Cold Failover Clusters, lecz dopuszcza równoczesną pracę wielu węzłów (rys. 5). Gdy wszystkie węzły są sprawne, wtedy wspólnie obsługują one żądania użytkowników. W przypadku awarii jednego węzła, kierowane do niego żądania są przejmowane przez inne węzły aktywne. W sytuacji awarii administrator nie wykonuje żadnych manualnych operacji.

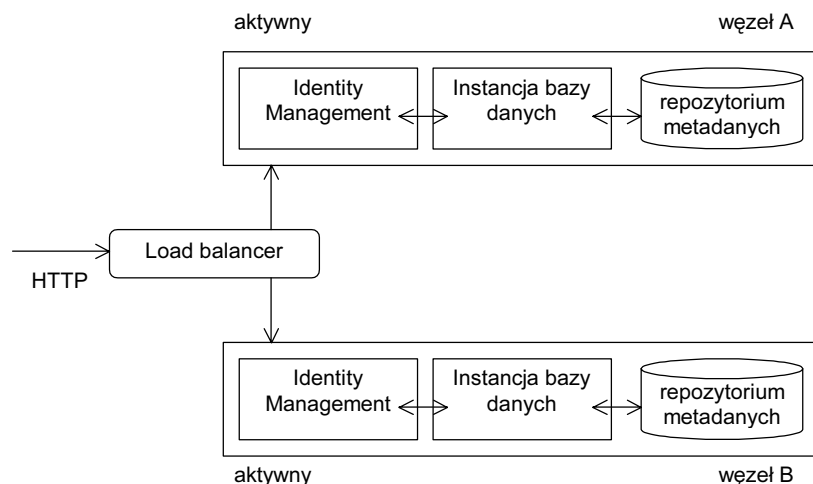
Konfiguracja Active Failover Cluster wymaga wykorzystania dwóch dodatkowych składników: modułu rozpraszania żądań (load balancer) oraz serwera bazy danych RAC (Real Application Clusters). Moduł rozpraszania żądań jest konieczny do kierowania żądań użytkowników na różne z aktywnych węzłów, a także do powstrzymania przekazywania żądań do węzła, który uległ awarii. Natomiast serwer bazy danych RAC musi zostać użyty ze względu na konieczność dostępu do plików jednej bazy danych z wielu równocześnie pracujących instancji bazy danych. Wszystkie serwery aplikacji, korzystające z infrastruktury, odwołują się w swoich konfiguracjach do adresu fizycznego modułu rozpraszania żądań (load balancer).



Rys. 5. Infrastruktura w architekturze Active Failover Cluster

#### 4.4.3. Replikacja usług zarządzania bezpieczeństwem

W tej architekturze każdy z węzłów posiada autonomiczną instalację komponentów IM i instancji bazy danych oraz własne repozytorium metadanych (rys. 6). W celu ujednoczenia zawartości repozytoriów metadanych stosuje się replikację. Serwery aplikacji przekazują swoje żądania za pośrednictwem modułu rozpraszania żądań (load balancer). Nie wymaga się stosowania serwera bazy danych RAC. W przypadku awarii jednego węzła, moduł rozpraszania żądań kieruje żądania użytkowników do innych aktywnych węzłów. W sytuacji awarii administrator nie wykonuje żadnych manualnych operacji.



Rys. 6. Infrastruktura w architekturze replikacji usług zarządzania bezpieczeństwem

## 5. Podsumowanie

Zapewnienie wysokiej niezawodności serwera aplikacji OAS 10g wymaga użycia sprzętowej redundancji polegającej na konstruowaniu trzech typów klastrów: klastrów serwerów aplikacji, klastrów serwerów buforujących WebCache i klastrów serwerów zarządzania bezpieczeństwem. W artykule omówiono podstawowe technologie zwielokrotniania komponentów OAS 10g oraz rozpraszania żądań użytkowników, umożliwiające budowę wymienionych konfiguracji.

## Literatura

1. Oracle Application Server 10g Documentation, Oracle Corp., <http://www.oracle.com/technology/documentation/appserver10g.html>