

XV Seminarium PLOUG
Warszawa
Maj 2007

Klastrowe architektury serwera aplikacji Oracle Application Server 10g

Mariusz Masewicz
Politechnika Poznańska

Mariusz.Masewicz@cs.put.poznan.pl

Abstrakt. Współczesny model przetwarzania danych, to skomplikowane aplikacje uruchamiane przy pomocy prostej przeglądarki internetowej działającej po stronie użytkownika z wykorzystaniem rozbudowanej architektury wielowarstwowych serwerów aplikacji. Architektura taka, choć z pozoru skomplikowana i trudna w zarządzaniu powinna w rzeczywistości pozwalać na łatwe jej skalowanie i zapewnianie niezawodności całego serwera, jak i poszczególnych jego komponentów. Nie bez znaczenia jest też możliwość łatwego zarządzania coraz to nowszymi wersjami udostępnianej aplikacji użytkownika. Zdecydowana większość serwerów aplikacji - oferowanych obecnie na rynku – w mniejszym, lub większym stopniu spełnia te założenia. Najpopularniejszym sposobem sprostania bardzo nieraz wygórowanym wymaganiom użytkowników (nieograniczona wręcz skalowalność, przy zagwarantowanej dostępności serwisu rzędu 99,99% czasu) jest umożliwienie tworzenia rozwiązań klastrowych. Serwer aplikacji Oracle Application Server 10g jest właśnie przykładem takiego rozwiązania, które pozwala na klastrowanie w zasadzie dowolnej warstwy oprogramowania.

1. Wstęp

Stosowana obecnie wielowarstwowa architektura większości programów informatycznych wymaga, aby fragmentem infrastruktury potrzebnej do ich obsługi był serwer aplikacji. Rynek oferuje wiele produktów tego typu, od darmowych (często bez niezbędnego wsparcia asysty technicznej, posiadających ograniczoną funkcjonalność, wystarczającą jednak być może dla małych, niewymagających aplikacji), po kompleksowe rozwiązania dużych, uznanych producentów (BEA WebLogic, IBM WebSphere, Oracle Application Server). W największym skrócie serwer aplikacji to zestaw usług (serwer HTTP, kontener aplikacji J2EE, infrastruktura zarządzania aplikacjami i użytkownikami i inne).

W miarę wzrostu wymagań stawianych wobec serwowanej aplikacji znaczenia nabierają takie czynniki, jak wspomniane wcześniej wsparcie techniczne, czy oferowane usługi dodatkowe (szyfrowanie, zapewnianie ciągłości i wydajności pracy).

Klastrowanie (z wielokrotnianie) poszczególnych składowych serwera jest jedną z najczęściej stosowanych metod spełniania wymagań użytkowników. Można wskazać wiele powodów, dla których klastrowanie różnych elementów infrastruktury informatycznej, a w szczególności serwerów aplikacji, jest coraz powszechniej stosowaną technologią. Do najważniejszych należą:

- zwiększenie niezawodności całego systemu – awaria pojedynczego węzła w klastrze nie powoduje przestoju serwera jako całości
- zwiększenie skalowalności – poprzez możliwość łatwego dodawania kolejnych elementów klastra
- atrakcyjna cena – budowę klastra można rozpocząć od zakupu niewielkiej ilości serwerów i później w miarę zwiększających się potrzeb, dokładać potrzebne serwery lub tylko komponenty serwera aplikacji

Dokumentacja serwera aplikacji Oracle Application Server 10g [OASha] wyszczególnia dwie architektury klastrów serwera aplikacji:

- aktywny-aktywny
- aktywny-pasywny

W obu tych architekturach poszczególne elementy serwera aplikacji są wielokrotniane tak, aby w sytuacji krytycznej zapewnić ciągłość pracy a w przypadku architektury aktywny-aktywny uzyskuje się także wzrost wydajności rozwiązania klastrowego w stosunku do instalacji na pojedynczym węźle.

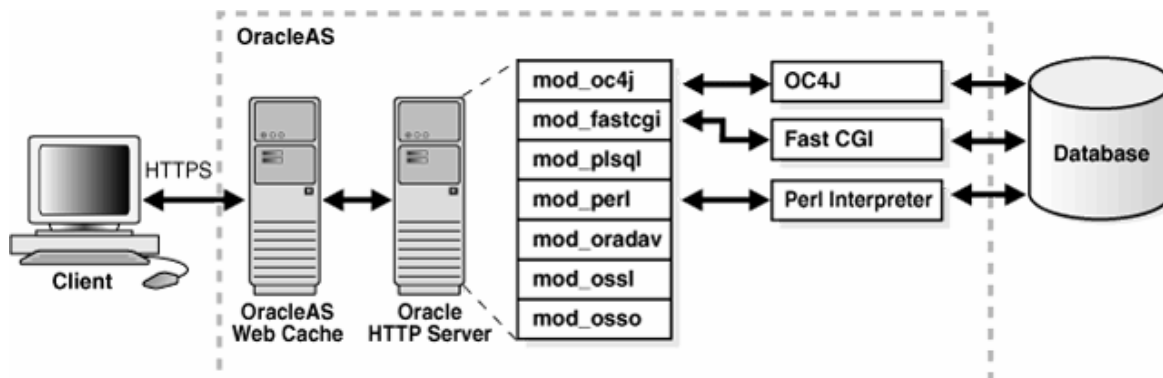
Należy zaznaczyć tu, że obie te architektury są tylko pewnego rodzaju propozycjami rozwiązań, gdyż Oracle Application Server 10g pozwala w ramach każdej z nich na praktycznie dowolne klastrowanie poszczególnych składowych serwera aplikacji.

Niniejszy artykuł omawia poszczególne aspekty klastrowania serwera aplikacji oraz poszczególnych jego składowych.

2. Architektura Oracle Application Server 10g

Serwer aplikacji jest to na ogół produkt, składający się z wielu niezależnych modułów, które współpracując ze sobą pozwalają na udostępnianie użytkownikom aplikacji wykonanych w różnych technologiach webowych (J2EE, PHP, PERL, ...). Bardzo często aplikacje te korzystają z baz danych jako repozytorium zarówno dla danych użytkownika, jak i metadanych niezbędnych do pracy serwera aplikacji. Na rysunku pierwszym widzimy typową architekturę serwera aplikacji (jest to uproszczony schemat architektury OAS 10g. Widzimy, że najważniejszą rolę w tym rozwiązaniu spełnia serwer HTTP (w przypadku OAS 10g jest to Apache). Serwer ten potrafi spełniać żądania użytkowników i w zależności od potrzeb uruchamiać aplikacje posługując się modułami (wtyczkami) odpowiednimi dla technologii, w której została wykonana aplikacja wywoływana przez użytkownika. Aplikacja taka bardzo często może sięgać do bazy (baz) danych.

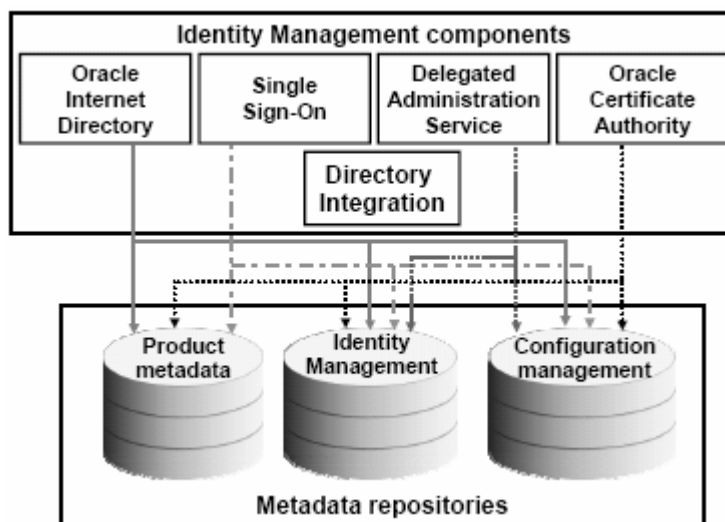
Bardzo często serwer HTTP jest wspierany przez usługę nazywaną tutaj Web Cache – czyli bufor (często posiadający rozbudowaną inteligencję), którego zadaniem jest filtrowanie żądań użytkownika i w zależności od tego, czy dotyczy ono wcześniej już wysyłanej strony (bądź jej fragmentu) – strona jest pobierana z bufora, a jeżeli strona jest żądana po raz pierwszy – żądanie jest przekazywane do serwera HTTP. Jest to jeden z mechanizmów zwiększania wydajności serwera aplikacji – poprzez odciążenie serwera HTTP i pozwolenie mu na wykonywanie tylko tych żądań, które są „nowe” (a więc „ciekawe”) w stosunku do poprzednio obsługiwanych żądań.



Rys. 1. Ogólna architektura serwera aplikacji (na przykładzie OAS 10g)

Oracle Application Server 10g posiada jednak bardziej rozbudowaną architekturę, w skład której wchodzi dwa rodzaje instalacji OAS 10g. Pierwszą z nich nazywamy Infrastruktura, a drugą właściwym Serwerem Aplikacji (AOS Middle Tier).

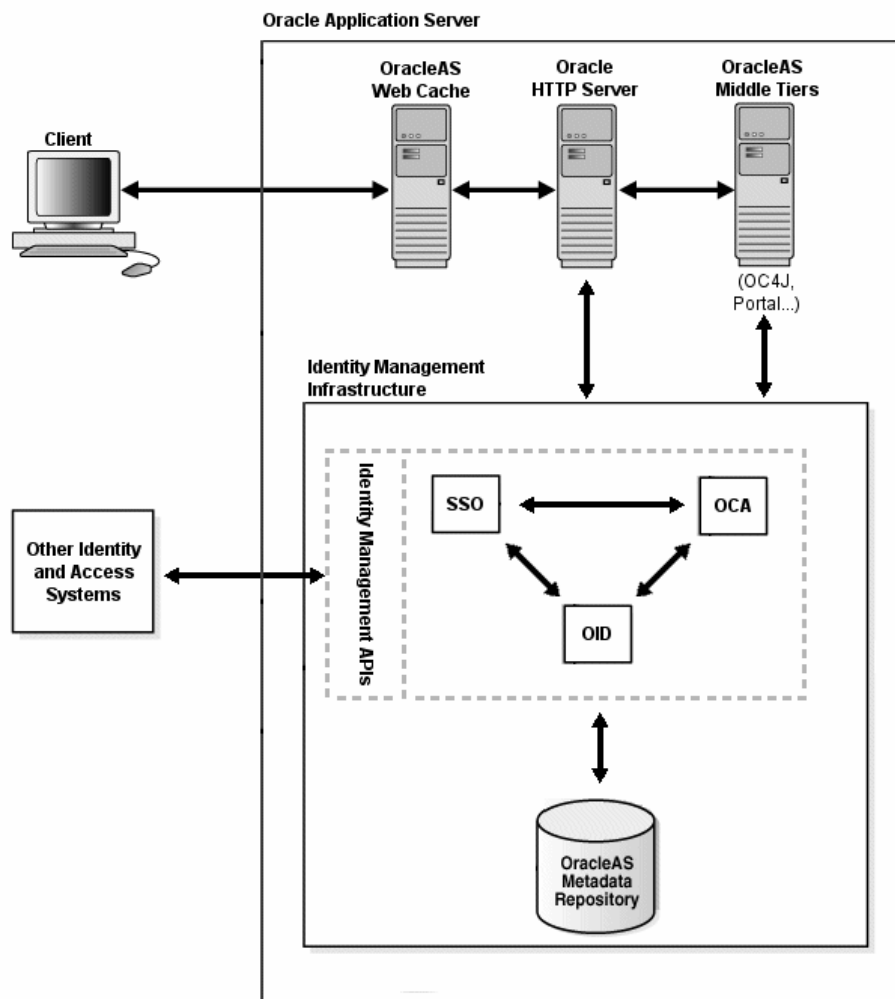
Zadaniem Infrastruktury jest dostarczenie szeregu usług związanych z zarządzaniem użytkownikami (katalog użytkowników, mechanizmy autentykacji i autoryzacji użytkowników), zarządzaniem serwerem aplikacji (repozytorium metadanych serwera aplikacji) i zarządzaniem aplikacjami działającymi w ramach serwera aplikacji (repozytorium metadanych aplikacji).



Rys. 2. Komponenty Infrastruktury serwera aplikacji OAS 10g

Drugim typem instalacji serwera aplikacji OAS 10g jest właściwy Serwer Aplikacji (do jego działania niezbędna jest jednak instalacja infrastruktury). W skład tej instalacji wchodzi serwer HTTP, usługi Web Cache, kontener OC4J (serwer aplikacji J2EE) i szereg usług dodatkowych takich jak Oracle Portal, serwer raportów, ...

Pełną architekturę serwera aplikacji OAS 10g przedstawia rysunek 3.



Rys. 3. Infrastruktura i Serwer Aplikacji OAS 10g

3. Wysoka wydajność i dostępność serwera aplikacji

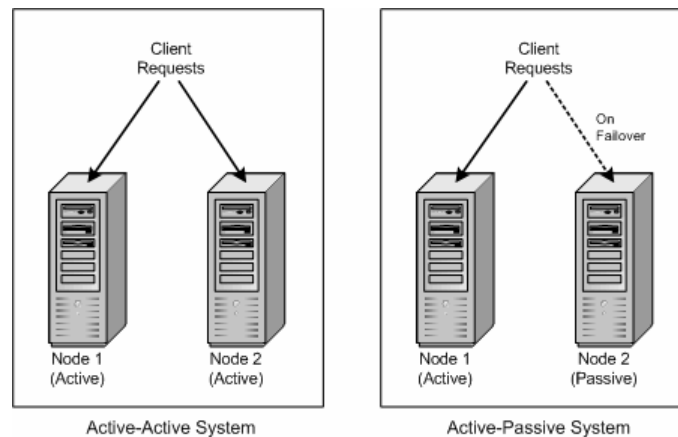
Dokumentacja serwera aplikacji OAS 10g pod hasłem „wysoka wydajność” definiuje pojęcie skalowalności tego produktu. Oznacza to, że poszczególne komponenty OAS 10g pozwalają na taką konfigurację, aby w razie wzrostu obciążenia tego elementu można było dołożyć kolejny element tego samego rodzaju i wówczas przejmie on obsługę części żądań trafiających do tej warstwy serwera aplikacji. Dokumentacja definiuje pojęcie skalowalności poziomej i pionowej. Pod terminem skalowalności poziomej rozumie się serwery aplikacji (lub ich komponenty) tworzące klastry – czyli odpowiednio skonfigurowane niezależne maszyny, których oprogramowanie współpracuje ze sobą dzieląc się obciążeniem. Skalowalność pionowa to możliwość uruchomienia kilku kopii tego samego komponentu (np. serwera HTTP) na tej samej maszynie – dzięki temu zmniejsza się obciążenie pojedynczego procesu działającego na danym serwerze.

Jako konfiguracje o podwyższonej niezawodności (dostępności) dokumentacja Oracle Application Server 10g wymienia klastry aktywny-aktywny i aktywny-pasywny.

Konfiguracja aktywny-aktywny to takie rozwiązanie, w którym można wydzielić przynajmniej dwie grupy (niekoniecznie identycznych) serwerów aplikacji, odpowiadających na żądania użyt-

kowników (często wspomagana mechanizmem równoważenia obciążenia). W razie awarii jednej grupy serwerów – pozostałe grupy są w stanie odpowiadać na żądania użytkowników (często z mniejszą wydajnością niż podczas normalnej pracy wszystkich elementów takiego klastra). Konfiguracja taka łączy w sobie cechy mechanizmów zapewniania wysokiej dostępności serwera z jego skalowalnością.

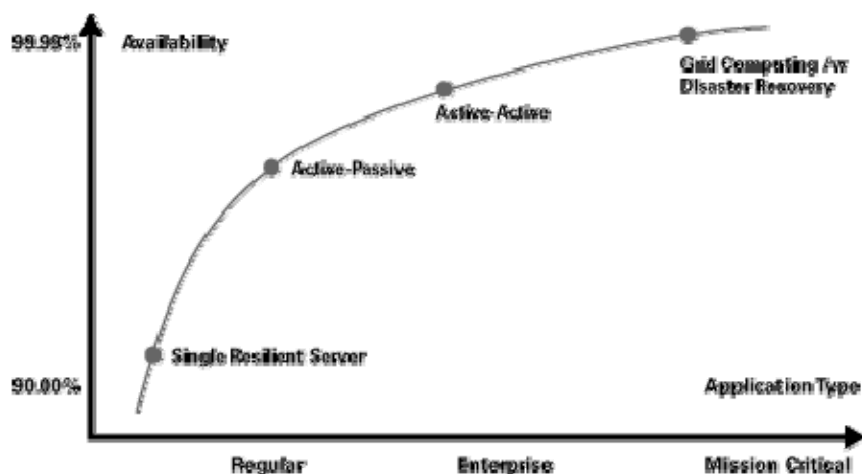
Klaster aktywny-pasywny to rozwiązanie, w którym na żądania użytkowników odpowiada jedna część serwerów wchodzących w skład klastra (serwery aktywne), podczas, gdy druga część serwerów pozostaje w uśpieniu. W przypadku wykrycia problemów z serwerami aktywnymi są one wyłączane a ich zadania przejmuje aktywowana (automatycznie lub ręcznie) grupa serwerów, które były do tej pory pasywne.



Rys. 4. Dwa rodzaje klastrów

OAS 10g pozwala na konfigurację usług w obu tych architekturach. Konfiguracje takie wymagają każdorazowo instalacji dodatkowych usług zarządzania klastrami (monitorowanie stanu klastra, zarządzanie wirtualnymi adresami serwerów, ...).

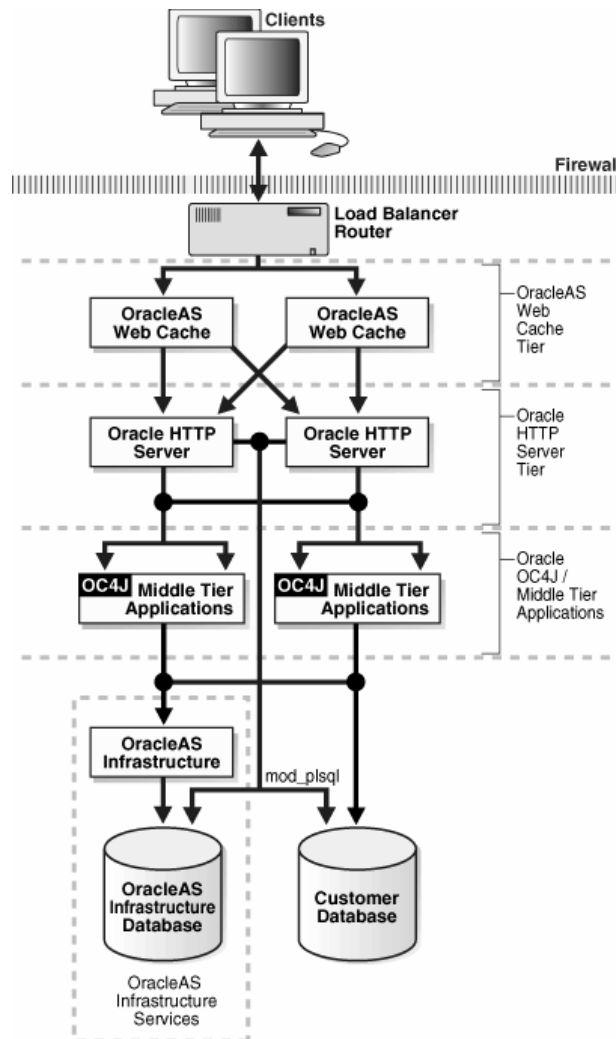
Oczywiście omówione wcześniej konfiguracje zawierają wskazówki na temat indywidualnej konfiguracji klastrów OAS 10g. W obu przypadkach dopuszczalne (a wręcz wskazane) jest klastrowanie poszczególnych komponentów serwera aplikacji.



Rys. 5. Zalecane rodzaje klastrów OAS 10g w zależności od planowanego wykorzystania serwera

4. Klastrowanie poszczególnych składników serwera OAS 10g

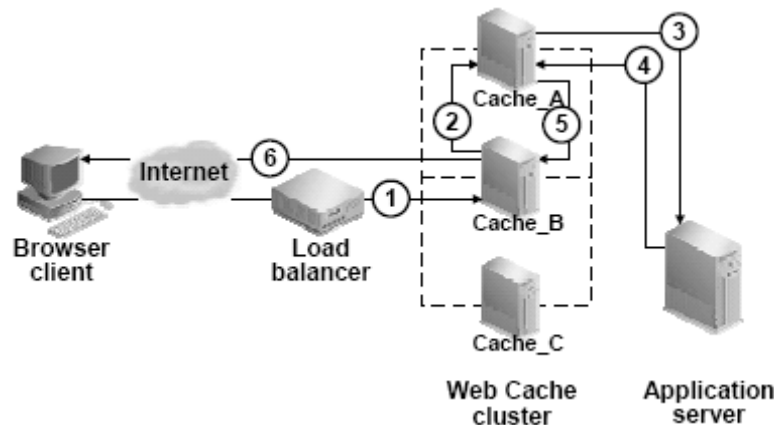
Jak już wspomniano wcześniej – architektura serwera aplikacyjnego Oracle Application Server 10g to zbiór modułów o jasno określonej roli. Dzięki temu możliwe jest określenie zapotrzebowania na usługi oferowane przez każdy z tych komponentów i dowolne zwielokrotnianie właśnie tych najbardziej obciążonych elementów. Dodatkowo mechanizmy zarządzania takimi klastrami komponentów (równoważnie obciążenia, zarządzanie rozproszonym repozytorium danej warstwy) sprawiają, że klastry te oferują faktyczny wzrost wydajności i niezawodności danego komponentu. Poniżej zostaną omówione mechanizmy klastrowania poszczególnych komponentów architektury OAS 10g.



Rys. 6. Klastry poszczególnych komponentów OAS 10g

Web Cache

Web Cache to usługa, której zadaniem jest odebranie żądania od użytkownika i ocena, czy zadanie takie przekazać do właściwego serwera HTTP, czy być może w repozytorium Web Cache już znajduje się aktualna wersja strony (lub jej części), o którą prosił użytkownik. W konfiguracji klastra serwerów Web Cache repozytorium tej usługi jest rozproszone – co oznacza, że wyszukiwanie buforowanej strony następuje we wszystkich repozytoriach serwerów Web Cache w klastrze. Przykładowy przepływ sterowania w takim wypadku pokazano na rysunku 7.



Rys. 7. Obsługa rozproszonego repozytorium serwerów Web Cache

Żądanie użytkownika (punkt 1) trafia do komponentu nazwanego Load Balancer (sprzętowy lub programowy mechanizm równoważenia obciążeń serwerów HTTP – lista licencjonowanych Load Balancerów każdorazowo znajduje się w dokumentacji do danej wersji serwera aplikacji OAS). Load Balancer wybiera jeden z serwerów Web Cache do obsługi żądania użytkownika. Jeżeli wybrany serwer nie posiada żądanej strony w swoim repozytorium – to wysła on zapytania (punkt 2) do pozostałych serwerów w klastrze Web Cache. W prezentowanym przypadku serwer A odpowiedział, że to on odpowiada za buforowanie żądanej strony. Z tego też powodu to serwer A poprosił serwer aplikacyjny o potwierdzenie aktualności lub wygenerowanie żądanej strony, po czym odesłał tę stronę do klienta za pośrednictwem obsługującego go serwera B. Serwer B przy okazji buforuje w swoim repozytorium stronę o którą prosił użytkownik.

W razie awarii jednego z serwerów Web Cache w klastrze ulega zgubieniu część repozytorium buforowana przez ten serwer – pozostałe fragmenty repozytorium są ciągle dostępne. Oznacza to, że tylko te strony, których zbuforowane kopie utracono będą podczas kolejnego żądania wyświetlane „znaczaco” wolniej.

W konfiguracji klastra serwerów Web Cache administrator określa znaczenie każdego z serwerów oraz ilość danych, które każdy z nich może zbuforować.

Klaster serwerów aplikacyjnych (HTTP)

Serwery HTTP (w przypadku Oracle Application Server 10g – Apache) pozwalają na uruchomienie kilku kopii podobnie skonfigurowanego serwera pracującego w klastrze. W przypadku OAS 10g zbiór serwerów aplikacyjnych nazywamy farmą. W ramach farmy może działać szereg instancji serwerów HTTP, współdzielących to samo repozytorium metadanych konfiguracyjnych (Infrastrukturę). Konfiguracja serwerów aplikacyjnych w farmie może znajdować się w plikach (wydzielone miejsce, dostępne dla wszystkich serwerów) lub bazie danych repozytorium metadanych infrastruktury. Każdy z tych serwerów może posiadać indywidualną konfigurację dotyczącą tego, jakie moduły i w jakiej ilości ma uruchamiać (liczba instancji OC4J, Oracle Portal, czy innych usług).

Klaster serwerów aplikacyjnych, to zbiór serwerów należących do tej samej farmy, wydzielonych do wykonywania tego samego zbioru aplikacji (i zarządzanych jako jedna całość – jednorazowa instalacja/modyfikacja aplikacji w ramach klastra, wspólny proces archiwizacji i odtwarzania).

Zadaniem serwerów HTTP jest przede wszystkim przekazanie żądania użytkownika do odpowiedniego modułu, w ramach którego uruchomiona jest aplikacja użytkownika (OC4J, PHP, PERL, PL/SQL,...). W przypadku klastrowania tej części serwera aplikacyjnego rolę mechanizmu Load Balancera może spełniać serwer (klaster serwerów) Web Cache lub w przypadku braku warstwy Web Cache (niezalecane) dowolny wspomniany wcześniej Load Balancer protokołu HTTP.

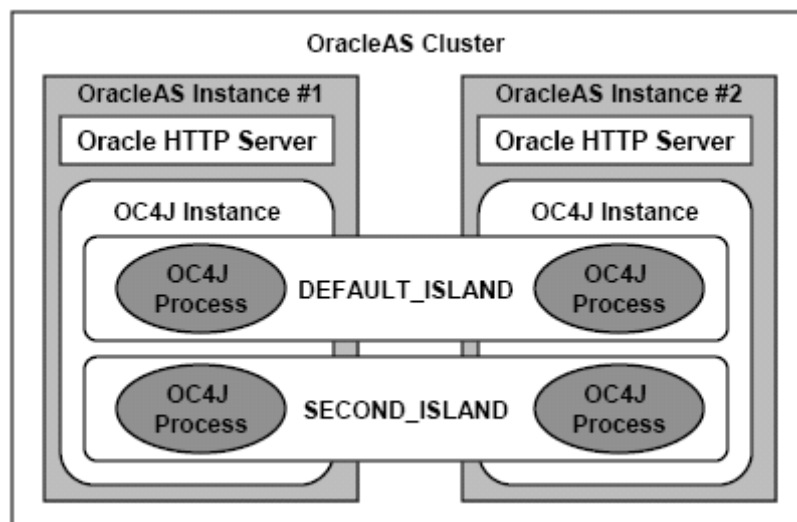
Klasy kontenerów OC4J

Kontenery OC4J są niezależnymi komponentami w ramach których wykonują się aplikacje u użytkownika stworzone w technologii J2EE. Kontenery OC4J pozwalają na uruchamianie różnych konfiguracji klastrowych. Stosowana tu terminologia to:

- instancja OC4J – jednostka w ramach której przygotowywane (wdrażane) są aplikacje J2EE. Jedna instancja może zawierać wiele procesów OC4J
- proces OC4J – wirtualna maszyna Javy, w ramach której działają aplikacje J2EE. Możliwe jest uruchamianie aplikacji w ramach dowolnego procesu należącego do instancji a także replikowanie stanu aplikacji pomiędzy procesami.
- wyspa OC4J – grupa procesów OC4J należących niekoniecznie do jednej instancji w ramach której uruchomiona jest logiczna kopia aplikacji J2EE (replikacji stanu aplikacji możliwa jest pomiędzy procesami w ramach tej samej wyspy). Aplikacja musi być skonfigurowana jako aplikacja rozproszona (można ją uruchamiać w ramach wielu procesów OC4J) i zachowująca stan (stan przetwarzania będzie replikowany w ramach wyspy), a za replikacji stanu aplikacji może odpowiadać na przykład mechanizm cookies przeglądarki użytkownika lub mechanizmy kontenera OC4J.

Rysunek 8 przedstawia typową architekturę klastra dwóch serwerów aplikacji. W ramach każdego z nich działa serwer HTTP i jedna instancja OC4J (w większości zastosowań uruchamia się więcej instancji). W ramach każdej instancji uruchomione są po dwa procesy OC4J, przy czym wyspy zostały wydzielone tak, że do każdej z nich należą po dwa procesy, ale należące do różnych instancji OC4J.

W przypadku, gdy serwer aplikacji przedstawiony na rysunku 8 otrzyma żądanie (otrzyma je jeden z serwerów HTTP – w zależności od tego, który zostanie wskazany przez zarządzającego takim klastrzem Load Balancera) wymagającego uruchomienia aplikacji J2EE – to serwer HTTP podejmuje decyzję, której instancji J2EE przekazać to żądanie. Następnie instancja określa, do której wyspy powinno to żądanie trafić i w ramach wyspy wybierany jest proces obsługujący żądanie. Administrator może określić, sposób przekazywania żądań do poszczególnych instancji, wysp i procesów OC4J (wybrać jeden z oferowanych algorytmów).



Rys. 8. Klasy serwerów aplikacyjnych (HTTP) i wyspy kontenerów OC4J

Klastrowanie baz danych

Większość aplikacji serwowanych w ramach serwera aplikacyjnego Oracle Application Server 10g korzysta z bazy danych jako składnicy danych aplikacyjnych (a sam OAS 10g wykorzystuje dodatkowo bazę danych jako repozytorium swoich metadanych).

Baza danych Oracle (we wszystkich stosowanych obecnie wersjach) oferuje szereg mechanizmów zapewniania wysokiej dostępności (wiele procesów nasłuchowych, Connection Manager) i wysokiej wydajności. Mechanizmem łączącym oba te wymagania jest tu znów klastrowanie (RAC, GRID, a także częściowo Data Guard). Naturalnym jest, więc że mechanizmy dostępu do baz danych stosowane w OAS 10g w pełni wykorzystują te mechanizmy. Dostęp do nich jest to możliwy poprzez prawidłową konfigurację modułu MOD_PLSQL, który może zarządzać pulami połączeń do bazy danych (w szczególności do różnych instancji w ramach klastra tej samej bazy danych).

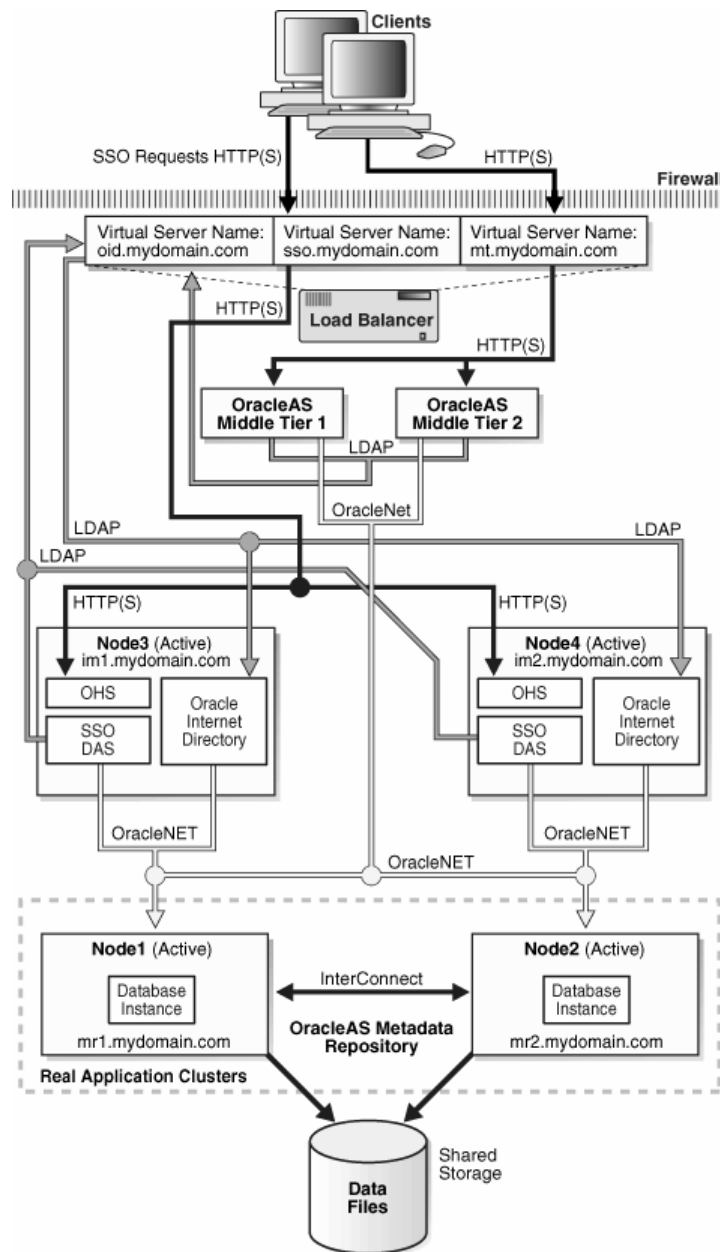
W przypadku rozwiązania opartego o bazodanowy klastr (RAC) można stosować szereg mechanizmów zapewniających ciągłość pracy użytkowników w razie awarii węzłów takiego klastra (np. przenoszenie stanu sesji użytkownika pomiędzy instancjami klastra).

Klastrowanie Infrastruktury

Także usługi Infrastruktury serwera aplikacyjnego Oracle Application Server 10g mogą podlegać klastrowaniu. W szczególności repozytorium metadanych – które jest składowane w bazie danych Oracle – podlega klastrowaniu zgodnie ze wszystkimi regułami klastrowania baz danych Oracle.

Usługi zarządzania tożsamością: Oracle Identity Management (Oracle Internet Directory, OracleAS Single Sign-On, Oracle Delegated Administration Services, oraz Oracle Directory Integration and Provisioning) mogą być uruchamiane zarówno w architekturze klastra aktywny-aktywny, jaki aktywny-pasywny. W pierwszej konfiguracji wymagane jest stosowanie mechanizmu równoważenia obciążeń – Load Balancera – także dla usług zarządzania tożsamością (protokół LDAP).

Oprócz rozwiązania przedstawionego na rysunku 9. (klastr warstwy Infrastruktury) możliwe są też inne konfiguracje, w których klastrowane są tylko usługi zarządzania tożsamością, bądź tylko baza repozytorium metadanych OAS 10g. Dokumentacja [OASha] w rozdziale „Oracle Application Server Infrastructure High Availability” szczegółowo przedstawia szereg różnych propozycji klastrów warstwy infrastruktury – zarówno w rozwiązaniu aktywny-aktywny, jak i aktywny-pasywny, zakładając instalację warstwy Infrastruktury, jak i serwera aplikacji na tych samych oraz różnych serwerach.



Rys. 9. Przykładowa architektura z klastrem w warstwie Infrastruktury

5. Podsumowanie

Serwer aplikacji Oracle Application Server 10g oferuje szereg mechanizmów pozwalających na praktycznie dowolne zarządzanie wydajnością i dostępnością serwowanych aplikacji. Omówione w niniejszym artykule przykłady klastrowania całego serwera jak i poszczególnych jego komponentów pozwalają na bardzo elastyczne określenie, które elementy i w jaki sposób mają oferować większą wydajność lub większą dostępność (lub obie te cechy razem). Dodatkowo łatwość konfiguracji i zarządzania tymi mechanizmami (z poziomu narzędzia Enterprise Manager) sprawiają, że administratorzy mogą bez zbędnego wysiłku spełnić wymagania stawiane wobec serwowanej aplikacji przez użytkowników końcowych.

Bibliografia

1. [OAS10g] <http://www.oracle.com/appserver/index.html>
2. [OASdoc] http://download-uk.oracle.com/docs/cd/B32110_01/index.htm
3. [OASha] http://download-uk.oracle.com/docs/cd/B32110_01/core.1013/b32201.pdf
4. [OASpg] http://download-uk.oracle.com/docs/cd/B32110_01/core.1013/b28942.pdf