ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: INFORMATYKA z. 34 Nr kol. 1381

Radosław PORCZYŃSKI
Politechnika Wrocławska, Instytut Sterowania i Techniki Systemów

WYBRANE ELEMENTY ARCHITEKTURY GLOBALNYCH SERWERÓW WWW

Streszczenie. Efektywna realizacja usługi WWW wymaga prawidłowego zaprojektowania architektury serwerów WWW.

W pracy przedstawiamy elementy procesu projektowania globalnych serwerów WWW służących obsłudze ogromnej ilości zapytań. Serwery te są tworzone zazwyczaj z wykorzystaniem wielu współpracujących ze sobą systemów komputerowych. W pracy skoncentrowano się na przedstawieniu zagadnień równoważenia obciążeń i pokazaniu trójwarstwowej architektury niezbędnej do realizacji takich systemów.

SELECTED ELEMENTS OF GLOBAL WWW SERVERS' ARCHITECTURE

Summary. Performance of WWW services requires special design of the architecture of WWW servers. This paper describes some crucial elements of global distributed WWW servers designing process. These servers are usually made of many cooperating computer systems. The main subject of this paper is WWW servers three-tier architecture used to balance the load of these servers.

1. Wstęp

Na świecie zainteresowanie usługami serwerów WWW wzrasta we wszystkich dziedzinach życia. Spełnienie potrzeb użytkowników wymaga prawidłowego zaprojektowania serwera WWW.

Wyróżnikiem serwera WWW jego jest wielkość, którą oceniamy według jednego z dwóch kryteriów:

- ilości zasobów udostępnianych użytkownikom;
- ilości zapytań generowanych przez klientów do konkretnego serwera.

Biorąc pod uwagę powyższe kryteria możemy określiś serwery WWW jako:

- *małe* - najczęściej prezentujące dane oddziału firmy, wydziału, zawierają około stu udostępnianych stron i uruchomione są na małych stacjach roboczych. Takie serwery obsługują po około pięć tysięcy zapytań dziennie.
- *średnie* – są to serwery prezentujące dane dużej firmy zawierają setki stron informacji, a pracują na serwerach dużych, posiadających zabezpieczenia na poziomie urządzeń wewnętrznych. Serwery te obsługują zwykle kilkadziesiąt tysięcy zgłoszeń dziennie.
- *duże* – serwery wykorzystywane przez dostawców usług internetowych lub firmy komputerowe, prezentujące dane w postaci kilkuset tysięcy stron. Pracują one na serwerach dużej mocy obliczeniowej z wykorzystaniem technik zabezpieczenia danych oraz pracy serwerów. Mogą to być np. serwery zawierające dane kilku departamentów firmy w jednym miejscu obsługujące około kilkaset tysięcy zgłoszeń dziennie.
- *globalne* – udostępniają ogromną ilość danych multimedialnych, posiadają kilka kopii jednego dokumentu w różnych miejscach, przykładem takiego serwera może być serwer WWW, który obsługiwał Letnię Olimpiadę w Atlancie w 1996 roku, a opracowany był przez firmę IBM[5]. Takie serwery obciążone są około milionem zapytań dziennie.

W szczególności serwery o charakterze globalnym muszą być tworzone z zachowaniem zasad prawidłowego projektowania. Prawidłowo przeprowadzony proces projektowania takiego serwera WWW wymaga rozróżnienia następujących elementów

- architektury sprzętu;
- zarządzania zadaniami odciążającymi serwer;
- zarządzania danymi na serwerze;
- analizy informacji o pracy serwera;
- zarządzania udostępnianiem informacji poprzez serwer WWW;
- sposobów przechowywania danych;
- bezpieczeństwa.

W Instytucie Sterowania i Techniki Systemów podjęte zostały prace nad nowymi algorytmami zarządzania zadaniami na serwerach WWW. Prace koncentrują się na opracowaniu nowej warstwy funkcjonalnej wchodzącej w skład architektury serwerów WWW - warstwy zarządzającej dostępem do serwera lub grupy serwerów.

W niniejszej pracy przybliżono wybrane zagadnienia procesu projektowania, które mają istotny wpływ na efektywne przetwarzanie zadań na serwerze, a w szczególności zajęto się równoważeniem obciążeń rozproszonych serwerów WWW.

Próba zarządzania zadaniami kierowanymi do serwera WWW może być podejściem do równoważenia obciążeń tych serwerów.

2. Architektura sprzętu

Przy wyborze sprzętu dla serwera WWW ważne jest, aby zdefiniować, które jego elementy mają największy wpływ na wydajność pracy.

Niestety nie można wprost zdefiniować wymagań dotyczących wymagań sprzętowych, a zależnych od wielkości serwera, czyli ilości kierowanych do niego zapytań. W dużym stopniu wymagania te zależne są od zastosowanej aplikacji realizującej serwer WWW.

Do budowy serwera WWW możemy wykorzystać każdy komputer, od małego PC po mainframe. Ważne jest jednak, aby posiadał on wystarczająco dobre narzędzia do zarządzania pracą systemu. Pozwoli to na skupienie uwagi na zarządzaniu serwerem WWW zamiast zajmowaniu się systemem.

W przypadku stosowania oddzielnych komputerów dla serwera WWW oraz serwera plików każdemu z nich stawiamy inne wymagania. Zazwyczaj serwer plików wymaga większej ilości pamięci oraz większej pojemności dysków, jak również stawiamy mniejsze wymagania co do szybkości pracy procesora. Każdy z serwerów wymaga jednak bardzo szybkiego podłączenia do sieci komputerowej. Jest to najbardziej krytyczny element serwera. Dla architektury wieloserwerowej najlepiej jest zachować homogeniczność serwerów. Pozwala to na łatwiejsze zarządzanie, a co za tym idzie skuteczniejszą realizację algorytmów równoważenia obciążeń.

2.1. Architektura sprzętu – pamięć masowa

W odniesieniu do pamięci masowej istnieją dwa podejścia. Jedno stawia na pierwszym miejscu czas dostępu do danych, a dopiero później ich nadmiarowość gwarantującą bezpieczeństwo. Drugie podchodzi do tego zagadnienia w sposób odwrotny.

Przy projektowaniu serwera należy wziąć pod uwagę zarówno czas dostępu, jak i stopień redundancji danych.

W przypadku małych serwerów lokalnie przechowujących dane preferowane jest zastosowanie najszybszej technologii dostępu do danych.

W przypadku gdy serwery WWW i serwer plików są niezależne, stosowany jest dwupoziomowy dostęp do danych.

Dane najczęściej używane przechowywane są lokalnie na serwerach WWW, a pozostała część dostarczana jest przez serwer plików (może być do tego zastosowany rozproszony

system plików - DFS serwer).W takim przypadku serwery plików nie muszą mieć zainstalowanych szybkich systemów dostępu do danych, powinny być jednak wyposażone w mechanizmy zabezpieczeń danych o dużej nadmiarowości.

2.2. Architektura sprzętu – infrastruktura lokalnej sieci komputerowej

W przypadku budowy serwera WWW zawierającego kilka różnych maszyn komunikacja pomiędzy nimi powinna być odseparowana od ruchu lokalnego generowanego przez innych użytkowników. Jest to szczególnie ważne, w przypadku gdy serwery WWW udostępniają dane w postaci audio lub wideo. Ważne jest również prawidłowe usytuowanie rutera IP oraz DNS serwera. Pozwoli to zrównoważyć obciążenie pomiędzy serwerami udostępniającymi dane. Zaleca się w takich przypadkach wykorzystywanie dynamicznego rutingu zamiast statycznych tablic. W większości przypadków ruch przychodzący do serwera WWW jest dużo mniejszy od ruchu wychodzącego, możemy tu zastosować rozwiązanie znane jako „thin pipe in, fat pipe out", wykorzystać wolniejszy standard LAN, poprzez który kierowane będą zapytania do serwera, a szybszy np. ATM lub FDDI jako droga wyjściowa z lokalnej sieci komputerowej do węzła sieci rozległej, dla danych generowanych przez serwer.

2.3. Architektura sprzętu – połączenie do sieci rozległej

Elementem, który bezwzględnie musi być włączony do rozważań przy opracowywaniu architektury serwera WWW, jest jego połączenie z siecią rozległą dającą komunikację z Internetem. Możemy do tego celu wykorzystać linie ISDN lub połączenia stałe.

Przy wyborze rozległej sieci powinniśmy zwrócić uwagę na następujące zagadnienia:

- szerokość pasma transmisyjnego;
- czas odpowiedzi serwera;
- bezpośrednie podłączenie do sieci;
- cyfrową transmisję;
- full duplex przy transmisji PPP;
- możliwość rekonfiguracji połączenia;
- dużą dostępność podłączenia.

2.4. Architektura sprzętu – archiwizacja danych

W przypadku archiwizacji danych serwerów WWW musimy zastosować inne podejście w porównaniu z pozostałymi systemami. Mamy tu do czynienia z systemem, który pracuje 24 godz. 7 dni w tygodniu, a dane przechowywane na nim zmieniają się w sposób dynamiczny, dlatego nie możemy sobie pozwolić na zatrzymanie jego pracy w celu przeprowadzenia

archiwizacji. Stosowanie narzędzi umożliwiających archiwizację otwartych plików w systemie bardzo ułatwi ten proces. W przypadku dużych serwerów lustrzanych dedykujemy jeden jako roboczy, a pozostałe posiadają dane tylko do odczytu. Ogranicza to konieczność przeprowadzania archiwizacji tylko do jednego serwera.

Dużą zaletą jest również posiadanie centralnego systemu archiwizacji danych. Upraszcza to w znacznym stopniu proces odtwarzania danych.

2.5. Architektura sprzętu – dostępność serwera

Stworzenie takiej infrastruktury serwera WWW wymaga podporządkowania się dwóm czasem wykluczającym się zasadom:

1. Izolować i eliminować każdą możliwość wystąpienia awarii.
2. Projektować system jak najprostszy.

Oznacza to, że struktura logiczna sieci powinna posiadać nadmiarowe połączenia do wszystkich ważnych elementów w niej zainstalowanych (tj. ruter, DNS, serwer plików). Jeżeli to tylko możliwe, zaleca się również posiadanie zdublowanych najważniejszych elementów komputera, np. karty sieciowej, dodatkowych dysków, sterowników dysków.

W przypadku rozproszonych serwerów lustrzanych wymagane jest dodatkowo, aby istniała możliwość:

- równomiernego rozkładu obciążenia pomiędzy serwery;
- obsłużenia zapytań kierowanych do uszkodzonego serwera przez serwer sprawny.

3. Równoważenie obciążenia pomiędzy serwery WWW

Jednym z najważniejszych parametrów określających popularność strony WWW jest ilość odwołań do tej strony. Duża ilość odwołań świadczy o dużej popularności informacji zawartych na danej stronie, jednakże powoduje to często znaczne pogorszenie jakości pracy z danymi. Duża liczba odwołań zwiększa obciążenie serwera oraz łącza wykorzystywanego do podłączenia serwera do Internetu.

Aby skrócić czas odpowiedzi serwera na zapytanie, w pierwszej kolejności możemy :

1. Zwiększyć przepustowość połączenia z internetem.
2. Zwiększyć szybkość pracy komputera, na którym uruchomiony jest serwer WWW (wymienić procesor, dodać pamięć, zainstalować szybsze dyski).
3. Przekazać część danych na drugi serwer (np. departamenty firmy mogą posiadać swoje własne serwery WWW na różnych komputerach).

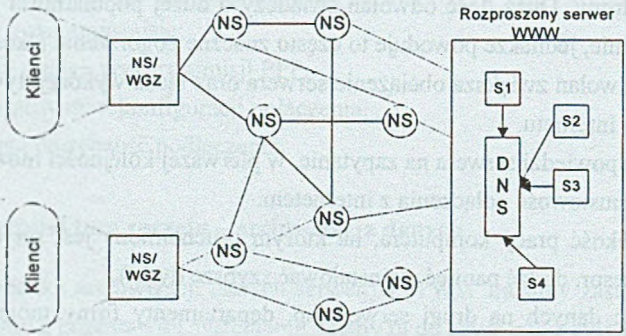
4. Utworzyć serwery lustrzane zawierające identyczne dane ze wzorcem umieszczonym w innym miejscu. Pozwala to użytkownikowi zdecydować, z którego serwera dane chce uzyskać.

Zaproponowane powyżej rozwiązania są idealne w przypadku serwerów oferujących niewielką ilość danych lub rozpoczynających pracę na niskim poziomie zainteresowania oferowanymi informacjami.

W IBM T.J. Watson Research Center zainteresowano się zagadnieniem równoważenia obciążeń serwerów WWW[1]. W rozproszonych systemach serwerów WWW, gdzie punkt centralny zarządza pracą pozostałych jednostek, zainteresowano się możliwościami wykorzystania DNS'a jako centralnego elementu sterującego obciążeniem pozostałych serwerów. Działanie serwera DNS (ang. Domain Name Server) polega na rozwijaniu zapytań tekstowych (logicznych) np. www.ces.pl na odpowiadający im adres IP (fizyczny) komputera w sieci. Zamiana nazwy komputera na jego numer IP może odbywać się etapami. Dla każdej domeny może istnieć oddzielny serwer nazw i aby uzyskać numer poszukiwanego komputera, należy przeszukać informacje na różnych serwerach. Dodatkowo lokalnie mogą istnieć serwery nazw NS, które zapamiętują pewne adresy IP i podają je bez sprawdzania w DNS. Lokalne serwery posiadają dodatkowo parametr określający, po jakim czasie dane przypisanie adresu IP do nazwy jest aktualne, a po którym muszą zapytać ponownie serwer DNS. Podjęto prace nad wprowadzeniem do algorytmów pracy serwera DNS elementów sterowania i równoważenia zapytań pomiędzy różne serwery WWW, a co za tym idzie równoważenia ich obciążenia. Rozważanie tej problematyki ma szczególne znaczenie dla wydajnej pracy globalnych serwerów WWW.

Zapytanie generowane przez klienta jest to żądanie dowolnego obiektu z serwera, np. tekstu, rysunku, dźwięku. Zapytania kierowane są do serwera WWW z wydzielonych węzłów zwanych dalej węzłem generującym zapytania.

Przy analizie przyjęto następujący model współpracy serwera DNS oraz WWW z siecią:



Rys.1. Model systemu

Fig. 1. System model

WGZ węzeł generujący zapytania;

DNS serwer nazw;

NS lokalny serwer nazw;

Si serwer WWW

W pierwszej kolejności podjęto prace nad algorytmami równoważenia obciążenia serwerów homogenicznych [2].

Udało się dostosować znane algorytmy dla efektywniejszego zarządzania pracą serwerów WWW. Są to algorytmy *Two tier Round Robin (RR2), Dynamically Accumulated Load (DAL), Dynamically Accumulated Load – set Actual Requests (DAL- AR)* [2]

Następnym etapem były prace nad algorytmami dla serwerów heterogenicznych. Zastosowano tu dodatkowo rozszerzenie poprzednich algorytmów o elementy wydajności względnej.

Zagadnienie równoważenia obciążenia pomiędzy serwery WWW możemy rozpatrywać w kontekście rozdziału zadań na poziomie adresów IP serwera WWW lub adresu fizycznego MAC komputera, na którym pracuje serwer WWW.

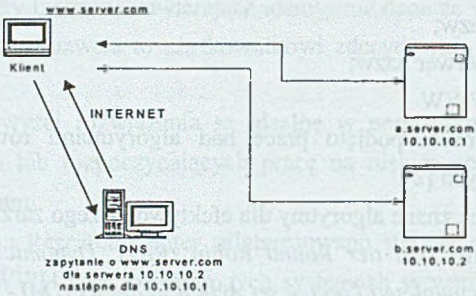
3.1. Równoważenie obciążenia pomiędzy serwery WWW – ruting adresów IP

W przypadku gdy mamy do czynienia z serwerami obciążonymi olbrzymią ilością odwołań do danych, problem rozdziału zadań pomiędzy serwery możemy rozwiązać korzystając z serwera DNS, który skieruje zapytanie do jednego z identycznych serwerów WWW posiadających swoją własną nazwę oraz adres IP.

Istnieje kilka strategii szeregowania zapytań realizowanych przez serwer DNS. Należy zwrócić uwagę na to, że różne strategie szeregowania będą pracowały efektywnie w przypadku serwerów homogenicznych (zbudowanych w oparciu o identyczne rozwiązania), a inne w przypadku serwerów heterogenicznych.

3.1.1. Algorytmy stosowane w przypadku serwerów homogenicznych

Działanie serwera DNS oparte jest na przypisywaniu nazwie słownej serwera odpowiedniego adresu IP. Nic nie stoi więc na przeszkodzie, aby serwer DNS przypisywał jednej nazwie różne adresy IP w zależności od przyjętego algorytmu. Rozwiązanie to byłoby idealne, gdyby nie istnienie lokalnych serwerów nazw NS, przechwytujących rozwinięcie nazwy na adres IP i przekazujących je przez określony czas TTL (ang. *time to live*) użytkownikom bez sprawdzania w docelowym serwerze DNS. Powoduje to w konsekwencji przeciążanie serwerów pomimo wykorzystania DNS'a do przekierowywania zapytań.



Rys. 2. Schemat działania DNS i strategii Round Robin

Fig. 2 Sheme of DNS and Round Robin strategies

Aby zmniejszyć wpływ takiego niekorzystnego zachowania się systemu, możemy zastosować jedną ze strategii: *Two tier Round Robin (RR2)* lub *Dynamically Accumulated Load (DAL)*.

Algorytm *Two tier Round Robin (RR2)* opiera się na zasadzie podziału na dwie grupy węzłów WGZ, poprzez które użytkownicy generują zapytania i przyporządkowaniu dla każdego innej wartości *TTL* oraz oddzielnym szeregowaniu zapytań do serwera. Daje nam to możliwość przewidzenia, przez jaki czas określony serwer będzie obciążony zapytaniami nie kontrolowanymi przez nasz serwer DNS, i skróceniu tego czasu do wartości nie powodującej przeciążenia serwera.

Innym rozwiązaniem może być algorytm zwany *Dynamically Accumulated Load (DAL)*[2]. W tym przypadku opieramy się na oszacowaniu obciążenia generowanego przez węzeł generujący zapytania użytkowników. W celu uzyskania prawidłowych wyników oszacowanie musi byś przeprowadzane dostatecznie często. Serwer DNS sumuje oszacowane obciążenie serwera WWW, do którego skierowane zostało zapytanie z konkretnego węzła. Daje nam to przewidywane zakumulowane (łącznie od początku pracy) obciążenie każdego serwera WWW. DNS wybiera więc serwer, który ten parametr ma najmniejszy.

W przypadku przedstawionych powyżej algorytmów nie mamy żadnej informacji zwrotnej od serwera WWW o jego aktualnym obciążeniu.

Zaproponowany algorytm DAL może zostać rozbudowany o kontrolowanie aktualnego stanu serwera WWW. Wystarczającą informacją jest binarny sygnał informujący o przekroczeniu przez serwer krytycznego poziomu obciążenia (może to być ilość zapytań, obciążenie procesora lub inna wartość, której przekroczenie powoduje pogorszenie jakości pracy serwera.) wysyłany asynchronicznie.

Stosując algorytm *Dynamically Accumulated Load – set Actual Requests (DAL- AR)* [2] strategia postępowania jest następująca. Po zakończeniu czasu określonego przez TTL serwer przesyła do DNS'a informacje o rzeczywistej obsłużonej ilości zgłoszeń. Pozwala to

zmniejszyć oszacowaną wartość zakumulowanego obciążenia serwera. W konsekwencji DNS wybiera nadal serwer z najmniejszą wartością zakumulowanego obciążenia.

3.1.2. Algorytmy stosowane w przypadku serwerów heterogenicznych

W przypadku serwerów heterogenicznych powstaje dodatkowy problem do rozwiązania, związany z tym, że różne serwery WWW mogą obsłużyć różną ilość zgłoszeń, nie powodującą ich przeciążenia. Analizując algorytmy wykorzystywane w przypadku serwerów homogenicznych można zauważyć, że wartość TTL przypisywana była wyłącznie na podstawie przewidywanego obciążenia generowanego przez węzeł. Stosując strategię przypisywania różnej wartości TTL w zależności od mocy obliczeniowej serwera oraz ilości generowanych zapytań możemy oczekiwać, że otrzymamy dużo lepsze rezultaty.

Algorytm probabilistyczny.[1] Załóżmy, że nasz serwer S_i charakteryzowany jest przez absolutną wydajność C_i oraz wydajność względną $s_i = (C_i/C_1)$ będącą stosunkiem wydajności do wydajności najlepszego serwera. Podczas wyboru serwera, do którego DNS kieruje zapytanie, generowana jest losowo liczba p ($0 < p \leq 1$). Wyboru serwera dokonujemy na podstawie porównania liczby losowej p oraz s_i dla serwera następnego w kolejności. Zapytanie jest kierowane do serwera tylko wtedy, gdy $p \leq s_i$. Wykorzystując powyższą strategię dla algorytmu *Round Robin*, gdzie z podziałem na grupy węzłów generujących różne obciążenia dla każdego węzła określamy wartość TTL. Załóżmy, że K to ilość grup, na które podzieliliśmy węzły generujące zapytania. Dla TTL/K niech TTL_j będzie wartością parametru TTL dla j-tego węzła generującego zapytania,

$$TTL_j = \frac{n}{t_j}$$

dla n będącego stałym parametrem, który skaluje całkowitą ilość przekierowań będącą:

$$n = \sum_{j=1}^{k} \frac{1}{TTL_j}$$

Mamy tu więc różne wartości parametru TTL w zależności od grupy, do której przypisaliśmy węzeł.

Algorytm deterministyczny.[1]. Pierwsza część algorytmu określająca strategię wyboru serwera obsługującego zgłoszenie jest taka sama, jak opisano powyżej, polega na wyborze serwera obsługującego zgłoszenie w zależności od jego wydajności. Zmianie ulega jedynie ilość i sposób obliczania wartości TTL powodujące zredukowanie liczby zapytań do serwerów słabszych, a zwiększenie ilości zapytań serwerów silniejszych. W tym przypadku definiujemy różne wartości parametru TTL dla grup węzłów, jak również różne dla różnych serwerów obsługujących zgłoszenia. Niech TTL_{ij} będzie wartością parametru TTL dla j-tego węzła i i-tego serwera, wtedy:

$$TTL_{ij} = \frac{n * s_i}{t_j}$$

Gdzie n jest parametrem określającym całkowitą ilość przekierowań adresów.

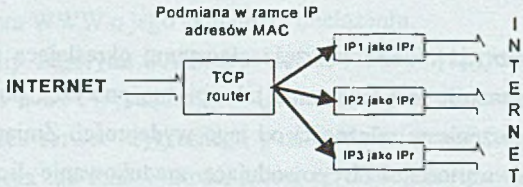
3.2. Równoważenie obciążenia pomiędzy serwery WWW – ruting adresów fizycznych MAC

Jak można zauważyć, DNS jest dobrym punktem na umieszczenie w nim mechanizmów zarządzających obciążeniem grupy serwerów, nawet w przypadku rozproszenia geograficznego. Daje on zadowalające efekty i jest stosowany w wielu miejscach na świecie.

W przypadku gdy mamy grupę serwerów i chcemy, aby zapytanie obsługiwał zawsze najmniej obciążony serwer, rozdział zapytań musimy przenieść na poziom adresów IP. Oznacza to stworzenie jednego serwera WWW z jednym adresem IP obsługiwanego przez kilka fizycznych komputerów. Aby uzyskać takie rozwiązanie, konieczne jest zastosowanie specjalnego oprogramowania na ruterze

Protokół IP umieszcza adres IP źródła i przeznaczenia w nagłówku , aby pakiet prawidłowo dotarł do celu. Używając dodatkowego oprogramowania nadajemy grupie serwerów ten sam adres IP. Posiadają one oczywiście również swoje własne adresy (IPn). Oprogramowanie postępuje tak umiejętnie, że po odebraniu pakietu z Internetu zmienia adres sprzętowy MAC na adres innego serwera i wysyła pakiet do sieci wewnętrznej. Odpowiedni serwer IPn odbiera pakiet i rozpoznaje adres IPr jako prawidłowe przypisanie. Serwer IPn obsługuje zapytanie i wysyła odpowiedź bezpośrednio do klienta korzystając z przypisania IPr podczas całego procesu komunikacji.

Protokół TCP jest zorientowany połączeniowo i dzieli informacje na pakiety, dodatkowo w transmitowanej ramce zapisywany jest numer portu TCP wykorzystywanego przez nadawcę i odbiorcę, aby określić, z którego połączenia pochodzi pakiet



Rys. 3. Schemat działania *software dispacher*
Fig. 3. Scheme of *software dispacher*

Tak więc oprogramowanie rutera musi mieć mechanizm umożliwiający zrealizowanie tego w prawidłowy sposób. Wykorzystywana jest do tego celu tablica przypisań:

<Client IP, Client PORT, IPr, Destination PORT> => IPn

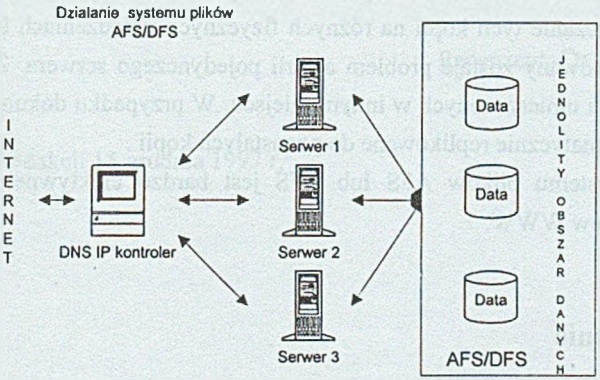
W oprogramowaniu tym może być zastosowana dowolna strategi wyboru serwera WWW obsługującego kolejne zgłoszenia. Nie ma również konieczności posiadania informacji zwrotnej od serwera WWW, ponieważ w pełni kontrolujemy ilość zgłoszeń do serwera.

4. Przechowywanie danych

Udostępniając informacje poprzez serwer WWW ważnym elementem, który należy rozpatrzyć, jest sposób składowania informacji. W większości przypadków są to informacje składowane w plikach na serwerze. W przypadku małych serwerów nie jest to problem, ale w momencie zwiększenia ilości zapytań do serwera może to być „wąskie gardło". W pierwszym momencie możemy serwer przenieść na komputer o większej wydajności, niestety rozwiąże to nasze problemy tylko na jakiś czas. W takich przypadkach należy zastanowić się nad konfiguracją wieloserwerową wyposażoną w mechanizmy równoważenia obciążeń.

W takim przypadku możemy wybrać jedno z dwóch rozwiązań:

- podzielić istniejącą strukturę katalogów z dokumentami pomiędzy kilka serwerów;
- utworzyć kilka kopii tego samego dokumentu i umieścić je na różnych serwerach.



Rys. 4. Dostęp do danych korzystając AFS/DFS
Fig. 4. Access to date using AFS/DFS

Pierwsze rozwiązanie wymaga od administratora zwrócenia szczególnej uwagi na spójność danych oraz powoduje zwiększoną ilość odwołań do różnych serwerów. Drugie z rozwiązań uważane jest za lepsze. Ważną rolę odgrywa tu jednak sposób dystrybucji

informacji pomiędzy serwerami. Można stosować do tego takie narzędzia jak ftp lub rdist. Jest to jednak metoda nieefektywna. Alternatywnym rozwiązaniem do tego jest wykorzystanie rozproszonego systemu plików. Do dyspozycji w zasadzie mamy trzy standardy: NFS (*Network File System*) firmy SUN, AFS (*Andrew File System*) firmy Transcarc oraz DFS (*Distributed File System*) opracowany przez Open Software Fundation (OSF). Wykorzystując system NFS każde odwołanie do danych generuje dodatkowy ruch w sieci. Może to spowodować sytuację, że zapytanie zostało skierowane do serwera o krótkim czasie odpowiedzi, a dane potrzebne klientowi znajdują się na serwerze o długim czasie odpowiedzi. W momencie wzrostu liczby zapytań może dojść do zablokowania niektórych serwerów.

Systemy plików AFS oraz DFS dostarczają inne rozwiązanie. Oparte jest ono na utworzeniu jednego drzewa plików identycznego dla każdego klienta. Pozwala to na dostęp serwera WWW poprzez klienta AFS'a lub DFS'a do danych bez konieczności znajomości fizycznego położenia plików. Każdy z serwerów WWW posiada lokalny system plików wykorzystywany jako cache przez klienta AFS'a lub DFS'a. Klient ten przechwytuje zapytania o dane od serwera WWW, sprawdza w obszarze lokalnym, a dopiero później wysyła zapytanie do serwera AFS'a lub DFS'a. W przypadku takiego rozwiązania lokalnie na komputerze obsługującym serwer WWW znajdują się dane najczęściej wykorzystywane i są one szybko dostępne dla klienta AFS'a lub DFS'a pracującego również na tym komputerze. Dobrze jest, by ten obszar był jak największy, aby zwiększyć współczynnik trafień. Dodatkowo systemy plików AFS oraz DFS oferują mechanizmy do replikacji danych. Proces ten umożliwia tworzenie wielu kopii danych tylko-do-odczytu (lub wolumenów). Pozwala to na umieszczanie tych kopii na różnych fizycznych urządzeniach lub serwerach. Dzięki temu wyeliminowany zostaje problem awarii pojedynczego serwera. Zawsze mamy dostęp do kopii danych umieszczonych w innym miejscu. W przypadku dokonywania zmian na danych są one automatycznie replikowane do pozostałych kopii .

Wykorzystanie systemu plików AFS lub DFS jest bardzo efektywne w przypadku rozproszonych serwerów WWW.

5. Podsumowanie

Przedstawione powyżej zagadnienia lokalizują problem równoważenia obciążeń w architekturze serwerów WWW. W rozproszonej architekturze serwera WWW wprowadzamy dodatkowy poziom zarządzania zapytaniami. W dalszym etapie naszych prac planujemy analizę i opracowanie nowych algorytmów równoważenia obciążeń. Ponadto nowe problemy pojawiają się w momencie przeniesienie zarządzania z punktu centralnego na rozproszone

węzły, na których bezpośrednio pracują serwery WWW. Pozwoli to na zwiększenie dostępności systemu poprzez wyeliminowanie jednego punktu centralnego odpowiedzialnego za zarządzanie całym systemem. Taka konstrukcja mechanizmu równoważenia obciążeń serwerów WWW wymaga dalszej analizy wydajności, jak również analizy zapytań kierowanych do serwera WWW.

LITERATURA

[1] Yu P. S., Colajanni M.: Dynamic Load Balancing in Geographically Distributed Heterogeneous Web Servers. Research Report RC 20741 IBM Research 2/20/97.

[2] Colajanni M., Yu P.S., Dias D.M.: Scheduling algorithms for distributed Web servers. submited to 17-th Int. Conf. On Distributed Computing Systems (ICDCS '97), Baltimore,MD, May 1997.

[3] Katz E.D., Butler M., .McGrath R.: A scalable HTTP server:the NCSA prototype. Computer Network and ISDN Systems. 1994, v. 27,s. 155-164.

[4] McGrath R.: What we do and don't know about the load on the NCSA WWW server. http://www.ncsa.uiuc.edu/InformationServers/Colloquia/28.Sep.94, Sierpień 1994

[5] Kay C., Vega G., Wong G.: How To Design and Manage an Enterprise Web Site. IBM Research, Advice Paper 04-11-96

Recenzent: Dr inż. Krzysztof Nałęcki

Wpłynęło do Redakcji 15 grudnia 1997 r.

Abstarct

In this work some elements of the process of designing of the WWW servers, that are being used to operate a huge amount of requests. Servers of that type are usually created with the usage of many co-operating together computer systems. In that work the main focus is put onto the requests of load balancing and displaying the three-tier architecture necessary for the realization of such processes.

A couple of strategies that can be used for server load balancing is presented here, by dynamic change of physical addresses that all answer to one logic address in DNS server. These are: *Two tier Round Robin, Dynamic Accumulated Load, Dynamic Accumulated Load set Actual Request.* Strategies presented are best checked in case of the homogenous servers WWW. For the heterogeneous servers additional modifications of the above strategies have been presented here. That additional modification equals the different performance coefficient of the server's.

It needs to be added that the above mentioned strategies have their weaknesses. The possibility of balancing of the server's WWW load balancing by using router IP for that. In case of using the dynamic change of the physical MAC station addresses the possibility of over-using the servers is eliminated and the requests are directed to the broken server.

In case of the distributed servers WWW providing a big amount of data some problems with their unity may arise. The usage of the distributed system of files AFS or DFS (Fig. 4.) may eliminate the problems connected with the unity of the data and correct the performance of the WWW server.