STUDIA INFORMATICA 2002
Volume 23 Number 3 (50)

Leszek BORZEMSKI
Politechnika Wrocławska, Instytut Sterowania i Techniki Systemów
Łukasz BŁASZKOWICZ
Politechnika Opolska, Wydział Elektrotechniki i Automatyki

WebDispatcher – SYSTEM RÓWNOWAŻENIA OBCIĄŻEŃ KLASTRA SERWERÓW WEBOWYCH

Streszczenie. W pracy przedstawiony został projekt nowego programowego przełącznika webowego WebDispatcher służącego do równoważenia obciążeń klastra serwerów webowych. WebDispatcher jest przełącznikiem zarazem warstwy czwartej, jak i siódmej, przeznaczonym do obsługi wieloserwerowych witryn WWW, dla których wybór serwera może być dokonany z wykorzystaniem statycznego algorytmu round-robin, dynamicznego algorytmu uwzględniającego aktualne obciążenie serwerów i kierującego żądanie do serwera najmniej obciążonego oraz algorytmu dokonującego dystrybucji żądań na podstawie typu żądanego obiektu. W pracy przedstawiono również wybrane wyniki przeprowadzonych pomiarów wydajności.

WebDispatcher – A LOAD BALANCING SYSTEM FOR WEB-SERVERS CLUSTERS

Summary. In this paper we present a new Web software switch design for supporting Web-servers clusters for high-performance and high-available WWW sites. This layer-4 and layer-7 switch provides load balancing to improve the site performance and has three dispatching policies options: static round-robin, server state-aware and content-aware policies. The design of the system was discussed as well as its implementation. The results of evaluation experiments are presented.

1. Wprowadzenie

Ostatnie lata rozwoju Internetu pokazują, że własności tej sieci odbiegają daleko od oczekiwań i wymagań stawianych przez użytkowników. W szczególności, dotyczy to jakości

usługi WWW, która stała się podstawową usługą Internetu. Badania pokazują, że w postrzeganym przez użytkowników opóźnieniu w dostępie do zasobów WWW prawie 40% udział mają niewydajnie funkcjonujące systemy komputerowe ośrodków webowych [4]. Osiąganie dużej wydajności było zawsze bardzo ważne przy projektowaniu systemów komputerowych. Ze względu na szybki rozwój technologii sprzętowych i programowych oraz gwałtowny przyrost liczby użytkowników Internetu i stawianych przez nich wymagań, trwałe osiągnięcie wysokiej wydajności systemów webowych stało się, jak nigdy dotąd, bardzo trudne do spełnienia. Podstawową architekturą systemu komputerowego ośrodka webowego, która umożliwia tworzenie rozwiązań spełniających stawiane postulaty, są systemy klastrów serwerów webowych, inaczej mówiąc klastry webowe. Należy dodać, że przed współczesnymi systemami internetowymi, obok wyzwania wydajności, stawiane jest wyzwanie wysokiej dostępności usług WWW, ich osiągalności oraz bezpieczeństwa, co w całości prowadzi do kompleksowego zdefiniowania wymagań w postaci tzw. wymagań jakości usług WWW QoWS [2]. Wśród prac prezentujących rozwiązania pozwalające polepszać jakość usług WWW należy wskazać na prace dotyczące jakości transferu sieciowego [5, 8, 10] oraz prace związane z jakością i wydajnością ośrodków webowych [1, 2, 3, 4]. Zagadnieniem tym interesuje się wielu producentów systemów sieciowych, którzy opracowali własne rozwiązania dystrybuujące ruch webowy [5, 6, 7]. Dwa z nich były już analizowane w ośrodku wrocławskim, a mianowicie programowy system IBM SecureWay Network Dispatcher [7] oraz sprzętowy przełącznik webowy Cisco Content Web Switch CSS 11050 [5].

Klastry serwerów webowych obsługują ruch pod kontrolą przełączników webowych, które realizują odpowiedni algorytm dystrybucji żądań użytkowników, ze względu na zadane kryterium wydajności. Najczęściej stosowanym kryterium jest wymaganie pracy klastra w warunkach zrównoważonego obciążenia serwerów. Jest to kryterium ważne z punktu widzenia operatora systemu. Inne kryteria dotyczą uzyskania pożądanego podziału obciążenia. Są też rozpatrywane kryteria, które uwzględniają wprost oczekiwania użytkowników końcowych, np. czas odpowiedzi [9]. Ostatnie kryterium jest trudniejsze do zaimplementowania w warunkach praktycznych aniżeli pozostałe kryteria, ale warto je rozpatrywać, bowiem lepiej niż kryteria równoważenia i podziału obciążenia oddaje oczekiwania użytkowników.

Niniejsza praca przedstawia autorskie rozwiązanie systemu równoważenia obciążeń klastra serwerów webowych, jakim jest system WebDispatcher. Projekt WebDispatcher jest odpowiedzią na zapotrzebowanie, które pojawiło się w trakcie dotychczas prowadzonych prac nad technologiami równoważenia obciążeń serwerów webowych. W trakcie prac

powstała potrzeba posiadania rozwiązania mogącego służyć jako platforma doświadczalna do implementacji i badań nowych algorytmów.

W kolejnych rozdziałach omawiane są główne założenia projektowe oraz przedstawiana jest architektura powstałego programowego przełącznika, będącego zarazem przełącznikiem warstwy czwartej, jak i siódmej. Zaprojektowany system jest przeznaczony do obsługi wieloserwerowych witryn WWW, dla których wybór serwera może być dokonany z wykorzystaniem statycznego algorytmu round-robin, dynamicznego algorytmu uwzględniającego aktualne obciążenie serwerów i kierującego żądanie do serwera najmniej obciążonego bądź algorytmu dokonującego dystrybucji żądań na podstawie typu żądanego obiektu. W pracy przedstawiono również krótko wyniki przeprowadzonych testów systemu.

2. Architektura systemu WebDispatcher

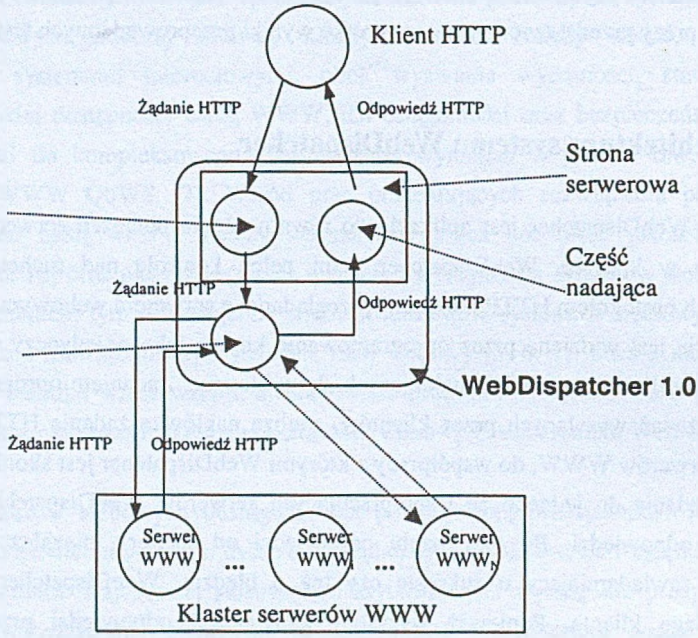
System WebDispatcher jest aplikacją do równoważenia obciążeń serwerów webowych pracujących w klastrze. WebDispatcher pełni pełną kontrolę nad ruchem żądań URL przesyłanych protokołem HTTP pomiędzy przeglądarką a serwerem webowym.

Aplikacja jest widziana przez oprogramowanie klienta jako pojedynczy serwer WWW, niezależnie od liczby serwerów znajdujących się w klastrze. Zadaniem oprogramowania jest odbieranie żądań wysyłanych przez klientów, analiza nagłówka żądania HTTP i rozsyłanie żądań do serwerów WWW, do współpracy z którymi WebDispatcher jest skonfigurowany. Po wysłaniu żądania do jednego ze współpracujących serwerów WebDispatcher oczekuje na komunikat odpowiedzi. Po otrzymaniu odpowiedzi od serwera, niezależnie czy jest to komunikat zawiadamiający o sukcesie czy też o błędzie, WebDispatcher odsyła ją do odpowiedniego klienta. Ponieważ wszystkie zapytania i odpowiedzi przechodzą przez WebDispatchera, to klienci nie mają pojęcia, od którego serwera tak naprawdę otrzymali odpowiedź. Z kolei serwery „myślą", że przez cały czas komunikuje się z nimi ten sam klient.

W programie WebDispatcher w wersji 1.0 zaimplementowano trzy algorytmy równoważenia obciążenia serwerów. Pierwszy z algorytmów to tzw. „karuzela" (ang. *round-robin*). Wówczas nowo przybyłe żądanie jest wysyłane do następnego (w stosunku do serwera, do którego zostało przekierowane żądanie poprzednio obsłużone) serwera z listy serwerów dostępnych. Algorytm ten nie bierze pod uwagę faktycznego obciążenia serwerów. Drugi algorytm wybierając serwer, do którego ma trafić nowo przybyłe żądanie, wyznacza decyzję na podstawie stanu liczników liczby aktywnych połączeń z serwerami. Algorytm ten bierze zatem pod uwagę faktyczne obciążenie (obciążenie serwera jest rozumiane jako liczba aktywnych połączeń) każdego z serwerów i wyśle żądanie do tego serwera, który jest, w danym momencie, najmniej obciążony. Trzeci i ostatni z zaimplementowanych w wersji 1.0

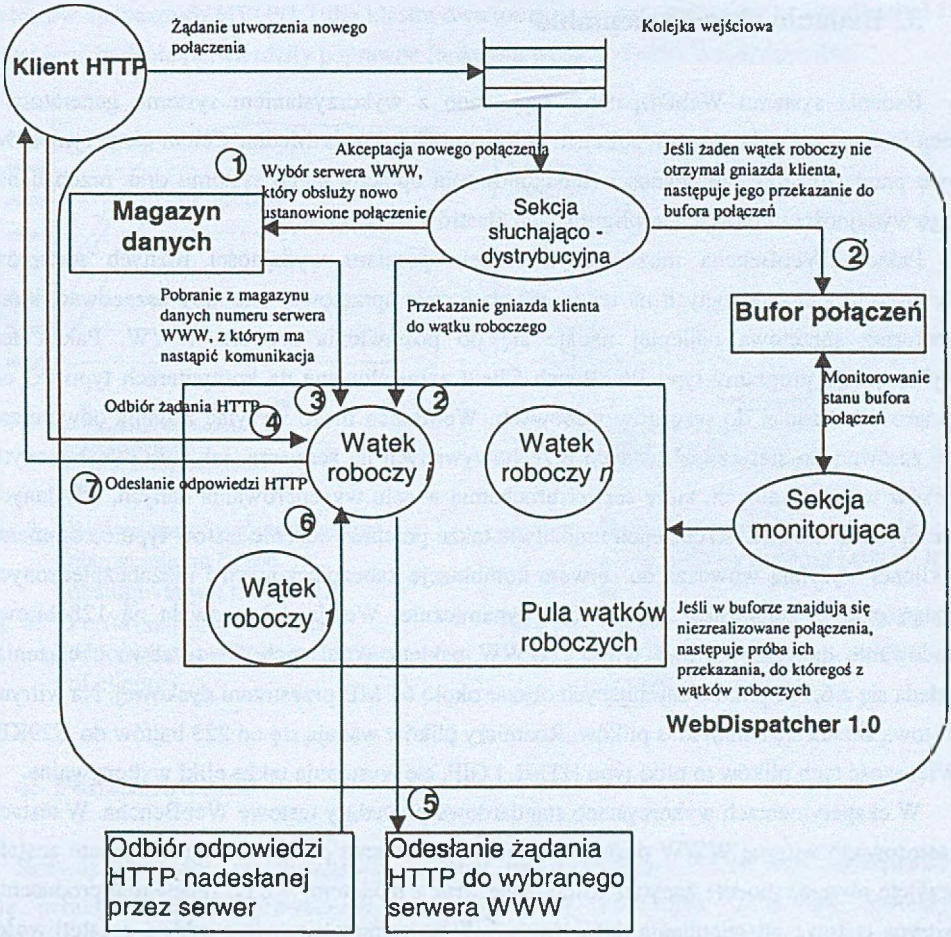
algorytmów poddaje szczegółowej analizie wiersz żądania HTTP i na podstawie typu żądanego przez klient obiektu dobiera odpowiedni serwer. Przykładowo, jeśli WebDispatcher współpracuje z dwoma serwerami, to żądania wszystkich typów tekstowych mogą być kierowane do jednego, a binarnych do drugiego serwera.

Na rysunku 1 przedstawiono organizację współpracy systemu WebDispatcher z klientami i serwerami. Jest to tzw. architektura dwukierunkowa [1]. Rysunek 2 przedstawia natomiast organizację wewnętrzną systemu WebDispatcher i kolejne kroki działań podejmowanych przez system podczas dystrybucji żądań.



Rys. 1. Dwukierunkowa architektura systemu WebDispatcher
Fig. 1. Two-way architecture of WebDispatcher

System WebDispatcher zaprojektowano w oparciu o architekturę wielowątkową, co umożliwia efektywne ustanawianie i obsługę wielu równoczesnych połączeń z klientami i serwerami WWW. Aplikacja jest gotowa do współpracy z klientami protokołu HTTP/1.0, ustanawiającymi wiele równoległych połączeń. Program umożliwia także ustanawianie trwałych połączeń protokołu HTTP/1.1. W testach, w których aplikacja pełniła rolę przełącznika protokołu HTTP/1.1, czasy utrzymywania połączeń z klientami i serwerami ustawione były na 5 sekund. Kiedy program pracował zgodnie ze standardem HTTP/1.0, opcja trzymania połączeń była wyłączona.



Rys. 2. Struktura systemu WebDispatcher
Fig. 2. Structure of WebDispatcher

Innym stałym parametrem konfiguracyjnym, niezależnym od rodzaju przeprowadzanego testu, była liczba wątków roboczych programu uruchamianych na starcie (40) oraz dopuszczalna liczba wątków roboczych (200). Wartości domyślne nie były zmieniane ze względu na to, że okazały się być wystarczające w każdym z przeprowadzanych testów. Niezmienna była także konfiguracja portu słuchającego WebDispatchera, który ustawiony był na domyślną dla WWW wartość 80. Inne parametry konfiguracyjne, takie jak: algorytm równoważenia obciążenia i liczba współpracujących serwerów zmieniały się w zależności od przeprowadzanego testu.

3. Badania eksperymentalne

Badania systemu WebDispatcher wykonano z wykorzystaniem systemu generatora i monitora ruchu webowego WebBench 4.1 firmy Ziff Davis Media. Celem eksperymentów było przetestowanie poprawności funkcjonowania opracowanego systemu oraz przebadanie jego wydajności w prostych konfiguracjach klastrów webowych.

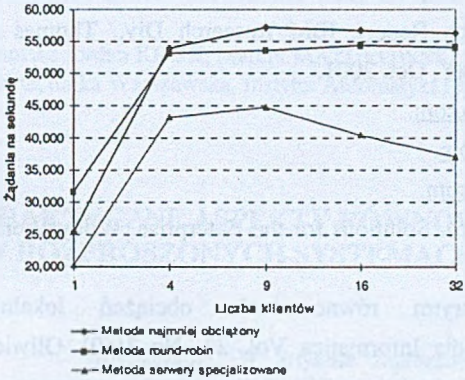
Pakietu WebBencha można użyć w celu pomiaru wydajności różnych serwerów webowych, zainstalowanych na tej samej platformie sprzętowej oraz aby oszacować, która platforma sprzętowa najlepiej nadaje się do postawienia serwera WWW. Pakiet ten wykorzystuje programy typu WebBench Client zainstalowane na komputerach typu PC do generowania żądań do serwerów webowych. WebBench może wysyłać żądania odwołujące się zarówno do statycznych danych przechowywanych na serwerze, jak i do dynamicznych plików wykonywalnych, które serwer uruchamia w celu wygenerowania danych, odsyłanych następnie do klienta. WebBench umożliwia także przeprowadzenie testów typu e-commerce - klienci wysyłają wówczas do serwera kombinację zabezpieczonych i niezabezpieczonych żądań o dane statyczne i generowane dynamicznie. Wersja 4.1 pozwala na 128–bitowe kodowanie danych. Testowa witryna WWW pakietu WebBench 4.1 (drzewo obciążenia) składa się z 6,160 plików zajmujących obszar około 61 MB przestrzeni dyskowej. Na witrynę testową składa się osiem klas plików. Rozmiary plików wahają się od 223 bajtów do 529KB. Większość tych plików to pliki typu HTML i GIF, ale występują także pliki wykonywalne.

W eksperymentach wykorzystano standardowe procedury testowe WebBencha. W testach zastosowano witrynę WWW dostarczoną przez producenta pakietu. Za producentem zostały przyjęte również modele zapytań dostarczone wraz z pakietem. Wedle zapewnień producenta witryna testowa odzwierciedla duże strony WWW, a modele zapytań oddają częstotliwość, z jaką obiekty każdej z klas wielkości są pobierane z popularnych witryn.

Wykonywano trzy grupy testów: (i) testy z żądaniami odwołującymi się do danych statycznych z użyciem protokołu HTTP/1.1 z trwałymi połączeniami, (II) testy z żądaniami odwołującymi się do danych statycznych z użyciem protokołu HTTP/1.0 oraz (iii) testy z żądaniami odwołującymi się do danych statycznych i dynamicznych plików wykonywalnych z użyciem protokołu HTTP/1.1 z trwałymi połączeniami.

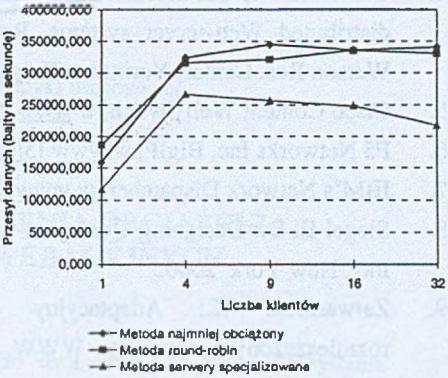
W badaniach wykorzystano proste konfiguracje klastrów webowych, budowane w wydzielonej sieci LAN zbudowanej na koncentratorze Ethernet Base-T, w skład której wchodziło 10 komputerów PC, z procesorem Intel Celeron 366MHz i 64MB pamięci RAM. Systemy wyposażone były w system operacyjny Microsoft Windows 98. Serwer webowy zbudowano w oparciu o program Microsoft Personal WebServer. Przykładowe wyniki badań

dla testów statycznych HTTP/1.1 dla klastra dwukomputerowego pokazano na rysunkach 3 i 4. Te i inne badania potwierdziły poprawne funkcjonowanie systemu WebDispatcher.



Rys. 3. Wpływ metody równoważenia obciążenia na liczbę obsługiwanych transakcji

Fig. 3. Performance of different distribution methods v. number of clients



Rys. 4. Wpływ metody równoważenia obciążenia na ilość przesyłanych danych

Fig. 4. Cluster throughput for different distribution methods v. number of clients

4. Podsumowanie

Rozwój systemów „uzdatniających" Internet jest gwałtowny. Wśród nich podstawową rolę pełnią systemy usprawniające pracę ośrodków webowych, a w tym systemy równoważenia obciążeń klastrów webowych. Do nich należy zaprezentowany w artykule system WebDispatcher. Badania eksperymentalne wykazały, że system funkcjonuje zgodnie z oczekiwaniami. System znajdzie zastosowanie jako element stanowiska badawczo-rozwojowego do badań nad nowymi algorytmami zarządzania systemami webowymi.

LITERATURA

1. Borzemski L., Nowak Z., Porczyński R.: Metody, algorytmy i rozwiązania systemowe równoważenia obciążeń serwerów WWW. Studia Informatica Vol. 21, No 1(39), Gliwice 2000.
2. Borzemski L.: Pomiary wydajności klastra serwerów WWW z równoważeniem obciążenia. Studia Informatica Vol. 22, No 1(43), Gliwice 2001.

3. Casalicchio E., Colajanni M.,: A client-aware dispatching algorithm for Web clusters providing multiple services. Proc. World Wide Web 10 , Hong Kong, May 2001.
4. Cardellini V., Casalicchio E., Colajanni M., Yu P.S.: The state of the art in locally distributed Web-server systems. Research Report, IBM Research Div., Thomas J. Watson Res. Center, Yorktown Heights, NY, Oct. 2001.
5. Cisco Content Web Switch, w www.cisco.com.
6. F5 Networks Inc. BigIP, w www.f5labs.com.
7. IBM's Network Dispatcher, w www.ibm.com.
8. Siegel E. D.: Designing Quality of Service. Solutions for the Enterprise. Wiley&Sons, Inc., New York 2000.
9. Zatwarnicki K.: Adaptacyjny algorytm równoważenia obciążeń lokalnie rozmieszczonych serwerów WWW. Studia Informatica Vol. 23, No 3(50), Gliwice 2002.
10. Zhao W., Olshefski D., Schulzrinne H.: Internet Quality of Service: an Overview. IEEE Network, September/October 1999.

Recenzent: Prof. dr hab. inż. Andrzej Grzywak

Wpłynęło do Redakcji 2 kwietnia 2002 r.

Abstract

We presented a new layer-4 and layer-7 Web software switch WebDispatcher designed for distributing user requests among a group of Web servers forming a Web cluster. Our switch supports three dispatching policies: round-robin, server state-aware and content-aware policies. The results of evaluation experiments were also presented.