STUDIA INFORMATICA 2003
Volume 24 Number 1 (52)

Henryk KRAWCZYK, Arkadiusz URBANIAK
Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki

STRATEGIE ZARZĄDZANIA KLASTEREM SERWERÓW WWW

Streszczenie. W pracy przedstawiono metody zwiększania efektywności funkcjonowania serwera WWW poprzez wykorzystanie idei obliczeń grupowych. Zaproponowano różne strategie zarządzania klasterem oraz przeanalizowano techniki równoważenia obciążenia. Rozpatrzono heurystyczne algorytmy rozdziału zapytań oraz dokonano oceny ich przydatności dla efektywnej realizacji usług WWW za pomocą metod symulacyjnych.

Słowa kluczowe: klaster, równoważenie obciążenia, serwer WWW.

WEB SERVER CLUSTER MANAGEMENT STRATEGIES

Summary. The method for improving efficiency of Web servers on the base of cluster computing is presented. Different cluster oriented management strategies are proposed and load balancing techniques are analysed. The heuristic algorithms of distribution of user requests are considered. Their suitability for efficient execution of Web services is evaluated by simulation approach.

Keywords: cluster, load balancing, web server.

1. Wprowadzenie

W ostatnich latach powszechny dostęp do Internetu spowodował znaczący wzrost obciążenia serwerów WWW. Liczba zapytań, adresowanych do serwerów WWW popularnych stron, rośnie w bardzo szybkim tempie. Z tego względu jednym z największych problemów jest wydajność serwerów WWW. Aby zwiększyć wydajność oferowanych serwisów, administratorzy coraz częściej tworzą grupy serwerów WWW, które dzielą między sobą realizowane zadania. Serwery te, połączone są ze sobą szybką siecią, stanowiąc klaster, który dla zwykłego użytkownika jest postrzegany jako jedna maszyna z jednym wirtualnym adresem IP. Na rynku informatycznym istnieje bardzo duża liczba narzędzi zarówno

programowych, jak i sprzętowych do tworzenia i zarządzania klasterem serwerów WWW. Poniżej podano reprezentatywne przykłady:

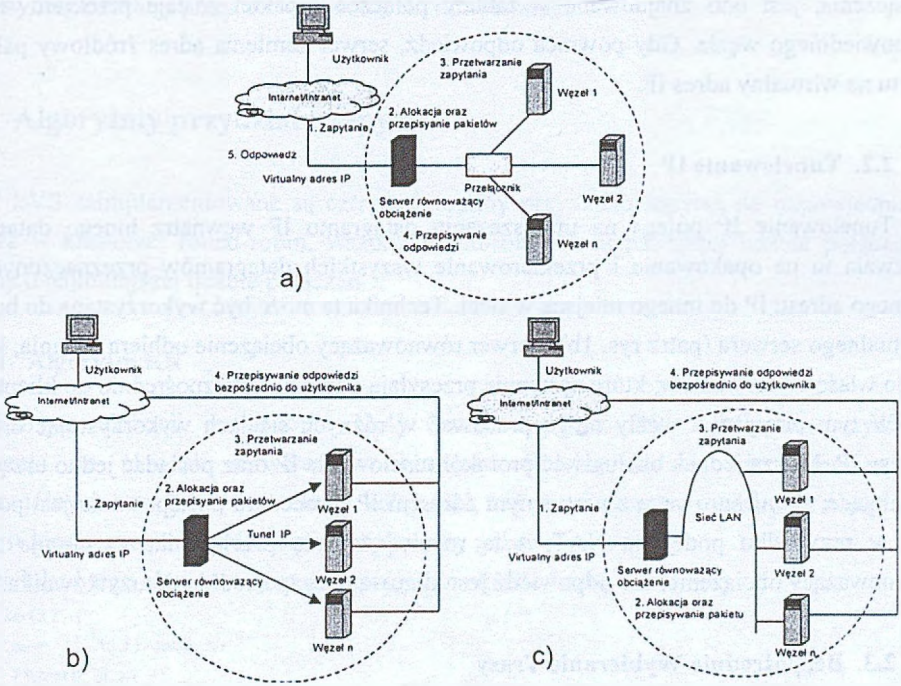
1) CISCO LocalDirector [1,6] – jest to jeden z pierwszych sprzętowych narzędzi do równoważenia obciążenia dostępnych na rynku. Działa on na zasadzie dystrybutora pakietów wykorzystując podwójne ich przepisywanie. Dystrybutor otrzymuje zgłoszenie od klienta, wybiera odpowiedni serwer WWW, który będzie odpowiedzialny za obsłużenie zgłoszenia i modyfikuje wszystkie nadchodzące pakiety zmieniając ich IP docelowe. Następnie, po wygenerowaniu przez serwer odpowiedzi, pakiety te trafiają ponownie do dystrybutora, który koryguje znów adresy IP w pakietach i odsyła je do klienta.
2) IBM SecureWay Network Dispatcher [1,7] – jest jednym z pierwszych komercyjnych programowych rozwiązań, które pozwala zwiększyć wydajność i zapewnić ciągłą pracę serwisów WWW. Sprowadza się również do dystrybutora pakietów, który przyjmuje pakiet od klienta i przekazuje go do serwera w klasterze używając fizycznego adresu MAC, bez konieczności jakiejkolwiek modyfikacji samego pakietu. Serwer WWW natomiast bezpośrednio odsyła pakiety do klienta bez dodatkowego obciążenia dystrybutora.
3) Linux Virtual Server (LVS) [3,4] - jest programowym narzędziem, które kieruje połączenia sieciowe do wielu serwerów dzielących obciążenie, mogącym służyć do budowy wydajnych i wysoko skalowalnych serwisów. Oprogramowanie LVS jest już wykorzystywane do budowy wielu serwisów o dużym obciążeniu w Internecie, takich jak portal Linuxa www.linux.com czy sourceforge.net. Działa on na zasadzie konfigurowalnego dystrybutora pakietów, który może pracować bądź jako dystrybutor z podwójnym przepisywaniem pakietów, bądź jako dystrybutor z przekazywaniem pakietów.

W dalszej części artykułu przeanalizowano różne techniki równoważenia obciążenia oraz zaprezentowano algorytmy przydziału zapytań wykorzystywane w LVS. W tym celu przedstawiono odpowiedni symulator i wykonano niezbędne eksperymenty umożliwiające porównanie wybranych algorytmów.

2. Techniki równoważenia obciążenia

Linux Virtual Server oferuje trzy różne techniki równoważenia obciążenia:

1) Sieciowa Translacja Adresów (ang. Network Address Translation, w skrócie NAT) [4].
2) Tunelowanie IP (ang. Tunneling IP, w skrócie TUN) [4],
3) Bezpośrednie Wybieranie Trasy (ang. Direct Routing, w skrócie DR) [2,4].



Rys. 1. Architektura LVS wykorzystująca podejście NAT (a), TUN (b) i DR (c)
Fig. 1. LVS architecture based on NAT (a), TUN (b) and DR (c) approach

2.1. Sieciowa Translacja Adresów

Z powodu niedoboru adresów IP w standardzie IPv4 coraz więcej sieci używa adresów prywatnych nie obsługiwanych przez routery Internetowe. Konieczność translacji adresów powstaje wtedy, gdy węzły w sieci wewnętrznej chcą uzyskać dostęp lub być dostępne z Internetu. Wykorzystuje się modyfikację adresów sieciowych zawartych w nagłówkach datagramu podczas ich przesyłania. Dzięki zastosowaniu prywatnych adresów sieciowych wymagany jest tylko jeden adres publiczny dla wirtualnego serwera zarządzającego.

Serwer równoważący obciążenie (patrz rys. 1a) jest połączony z węzłami za pomocą przełącznika. Gdy użytkownik chce uzyskać dostęp do usługi, wysyła pakiet do serwera równoważącego obciążenie o wirtualnym adresie IP. Serwer ten sprawdza adres przeznaczenia oraz numer portu. Jeśli są one zgodne z tablicą reguł wirtualnego serwera, to za pomocą odpowiedniego algorytmu alokacji wybierany jest węzeł, który obsłuży żądanie. Wówczas określone połączenie jest dodawane do tablicy zawierającej wszystkie aktywne połączenia. Adres przeznaczenia oraz port są zamieniane na adresy wybranego węzła i pakiet

zostaje do niego przesłany. Kiedy przychodzi pakiet należący do ustalonego już wcześniej połączenia, jest ono znajdowane w tablicy połączeń i pakiet zostaje przekierowany do odpowiedniego węzła. Gdy powraca odpowiedź, serwer zamienia adres źródłowy pakietu i portu na wirtualny adres IP.

2.2. Tunelowanie IP

Tunelowanie IP polega na umieszczaniu datagramu IP wewnątrz innego datagramu. Pozwala to na opakowanie i przekierowanie wszystkich datagramów przeznaczonych dla jednego adresu IP do innego miejsca w sieci. Technika ta może być wykorzystana do budowy wirtualnego serwera (patrz rys. 1b). Serwer równoważący obciążenie odbiera żądania, kieruje je do właściwych węzłów, które następnie przesyłają odpowiedź bezpośrednio do klienta.

W tym przypadku węzły mogą pracować w różnych sieciach wykorzystując dowolne adresy IP. Muszą jednak obsługiwać protokół tunelowania IP oraz posiadać jedno urządzenie tunelujące, skonfigurowane z wirtualnym adresem IP. Procedura postępowania jest podobna jak w przypadku podejścia NAT, z tą różnicą że rolę przełącznika przejmuje serwer równoważący obciążenie, zaś odpowiedź jest kierowana bezpośrednio do użytkownika.

2.3. Bezpośrednie Wybieranie Trasy

Serwer równoważący obciążenie oraz węzły usługowe muszą mieć jeden fizyczny interfejs dołączony za pomocą koncentratora lub przełącznika do stałego segmentu sieci LAN (patrz rys. 1c). Wirtualny adres jest dzielony przez węzły oraz ten serwer. Serwer ten przejmuje przychodzące zadania, które przekierowuje następnie do właściwego węzła. Wszystkie węzły usługowe mają wspólne dowiązanie do interfejsu, skonfigurowane na wykorzystanie wirtualnego adresu IP.

Klaster ten funkcjonuje w sposób podobny jak klaster wykorzystujący sieciową translację adresów czy tunelowanie IP. Podstawowa różnica to zmiana adresu MAC na właściwy dla danego serwera i przesłanie ramki do sieci LAN.

Sieciowa translacja adresów, pomimo że jest najprostszą metodą do zaimplementowania. posiada wielką wadę, jaką jest ograniczenie liczby serwerów węzłowych do około 20. Wiąże się to z tym, że serwer musi obsłużyć zarówno pakiety przychodzące, jak i wychodzące z systemu, a przy dużej liczbie tych pakietów może się stać wąskim gardłem całego systemu. Bezpośrednie wybieranie trasy, podobnie jak Tunelowanie IP, pozwala na pominięcie serwera równoważącego obciążenie dla przesłania wyników. To zapewnia zwiększenie wydajności systemu dzięki znacznemu zredukowaniu natężenia ruchu przechodzącego przez serwer równoważący obciążenie i ograniczeniu przepisywania pakietów. Dlatego też serwer równoważący obciążenie może obsługiwać ponad 100 węzłów i to nie powoduje efektów

wąskiego gardła sytemu. Wadą techniki DR jest skomplikowana konfiguracja, natomiast TUN wymaga obsługi przez serwer protokołu tunelowania.

3. Algorytmy przydziału zapytań

W LVS zaimplementowane są cztery algorytmy przydziału zapytań do odpowiedniego serwera w klasterze: round-robin, ważony round-robin, o najmniejszej liczbie połączeń i ważony o najmniejszej liczbie połączeń.

3.1. Algorytm RR

Algorytm alokacji round-robin [3,4] (oznaczony jako RR) cyklicznie przydziela połączenia do różnych serwerów w klasterze. Kolejne żądania kierowane są do poszczególnych serwerów w ustalonej kolejności, zaś po wyczerpaniu całej puli następne z nich jest kierowane ponownie na serwer pierwszy z listy.

Pseudokod tego algorytmu jest następujący (*n* określa liczbę serwerów):

```
while(1) {
    i = (i + 1) mod n;
    return S,;}
```

3.2. Algorytm WRR

Algorytm alokacji ważony round-robin [3,4] (oznaczony jako WRR) kieruje połączenie do serwera w klasterze w oparciu o aktualną przepustowość reprezentowaną przez odpowiednią wagę. Szeregowanie w WRR działa w następujący sposób:

Niech w klasterze znajduje się *n* serwerów, tj. $S = \{S_0, S_1, ..., S_{n-1}\}$, index *i* jest ostatnio wybranym serwerem z *S*, zmienna *cw* zaś bieżącą wartością wagi. Zmienna *i* i *cw* jest na początku równa zero. Jeżeli wszystkie serwery posiadają wagę $W(S_i) = 0$, to nie ma dostępnych serwerów do wykonania zadań i wszystkie połączenia do klastera (serwera wirtualnego) są odrzucane. Taką funkcjonalność określa poniższy pseudokod:

```
while (1) {
    if (i == 0) {
    cw = cw -1;
    if (cw <= 0) {
ustawiamy cw na największą wagę z S;
    if (cw == 0) return NULL;}
    } else i = (i + 1) mod n;
    if (W(S ) >= cw)
    return S ;}
```

W metodzie WRR wszystkie serwery z wyższą wagą jako pierwsze uzyskują nowe połączenie i dostają większą liczbę zadań niż serwery z niższą wagą. Serwery z równą wagą mają w przybliżeniu jednakowo rozproszone zapytania.

3.3. Algorytm NLP

Algorytm alokacji według najmniejszej liczby połączeń [3,4] (w skrócie NLP) kieruje połączenia do serwera, który posiada najmniejszą liczbę aktywnych połączeń. Wymaga więc zliczenia liczby aktywnych połączeń do każdego serwera na bieżąco. Dla klasterów zbudowanych z serwerów o podobnych osiągach metoda ta prowadzi do płynnego rozdzielania zapytań, nawet jeżeli obciążenie dla pojedynczego zapytania znacznie się różni.

Pseudokod tego algorytmu jest następujący:

```
while(1) {
    for(j = 0; j < n; j++) {
    pobieramy C_j liczbę aktywnych połączeń;
    if (C_m < 0 || C_m > C_j) {
    C_m = C_j;
    i = j;}}
    return S_i;}
```

3.4. Algorytm WNLP

Algorytm alokacji według ważonej najmniejszej liczby połączeń [3,4] (w skrócie WNLP) jest ulepszeniem algorytmu NLP, w którym waga może być przypisana do każdego serwera. Serwer posiadający większą wartość wagi będzie otrzymywał większy procent aktywnych połączeń w tym samym czasie. Administrator wirtualnego serwera może określać wagę do każdego serwera, i połączenia sieciowe będą przydzielane do każdego serwera w zależności od obecnej liczby aktywnych połączeń do każdego serwera i jego wagi. Niech każdy z serwerów ma przypisaną wagę W_i $(i=1,...,n)$ i aktywne połączenia C_i $(i=1,...,n)$, liczba wszystkich połączeń P jest sumą C_i $(i=1,...,n)$, połączenia sieciowe będą kierowane do serwera j, w którym: $(C_j/P)/W_j = min\ \{(C_i/P)/W_i\}$ $(i=1,...,n)$. W przypadku gdy P jest stałą, wzór ten można przedstawić następująco: $C_j/W_j = min\ \{C_i/W_i\}$ $(i=1,...,n)$.

Pseudokod algorytmu WNLP jest następujący:

```
while(1) {
    for(j = 0; j < n; j++) {
    pobieramy C_j liczbę aktywnych połączeń;
    pobieramy W_j wagę aktualnego serwera;
    CW_j = C_j / W_j;
    if (CW_m < 0 || CW_m > CW_j) {
    CW_m = CW_j;
    i = j;}
    } return S_i;}
```

4. Metoda oceny algorytmów

Do celów analizy technik równoważenia obciążenia i porównania różnych algorytmów alokacji połączeń został zbudowany symulator [5], którego zadaniem jest symulacja pracy klastera węzłów WWW. Poniżej przeanalizujemy podejście LVS z sieciową translacją adresów NAT. Ograniczymy się do zbadania dwóch algorytmów alokacji: RR i NLP.

Podstawowym kryterium oceny każdego z algorytmów równoważących obciążenie serwerów w klasterze jest średnia długość kolejki zadań oczekujących na wykonanie na poszczególnych serwerach. Dzięki temu można określić, czy serwery są równomiernie obciążone. Drugim parametrem jest czas oczekiwania klienta na odpowiedź na wysłane żądanie, co w dużym stopniu obrazuje wydajność całego systemu.

4.1. Eksperymenty i wyniki

W celu oceny wydajności wybranych algorytmów alokacji żądań przebadano trzy następujące scenariusze testowe.

Tabela 1

Parametry scenariuszy testowych

Scenariusz testowy	Liczba serwerów	Wydajność serwerów	Liczba klientów	Intensywność zgłoszeń	Wielkość danych
1	3	2 2 2	30	3 zadania/klienta	2 pakiety/zadanie
2	3	2 3 4			
3	2	2 2			

W tabelach (tabela 2, 3) i na kolejnych rysunkach (rys. 2, 3) przedstawiono wyniki symulacji.

Tabela 2

Czas oczekiwania klienta na odpowiedź systemu dla różnych scenariuszy testowych

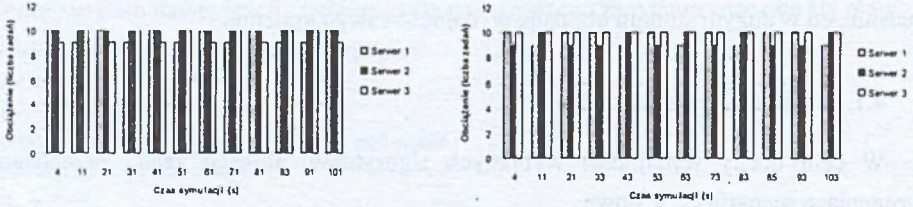
Scenariusz testowy	Zastosowany algorytm	Czas oczekiwania na odpowiedź (s)		
		Średni	Minimalny	Maksymalny
1	RR	30,21	12,29	40,20
	NLP	30,20	12,28	40,19
2	RR	38,13	20,50	92,42
	NLP	31,79	15,37	59,02

Tabela 3

Czasy odpowiedzi w funkcji liczby serwerów klastera

Liczba serwerów w klasterze	Czas oczekiwania na odpowiedź (s)		
	Średni	Minimalny	Maksymalny
dwa serwery	41,27	17,98	60,30
trzy serwery	30,20	12,28	40,19

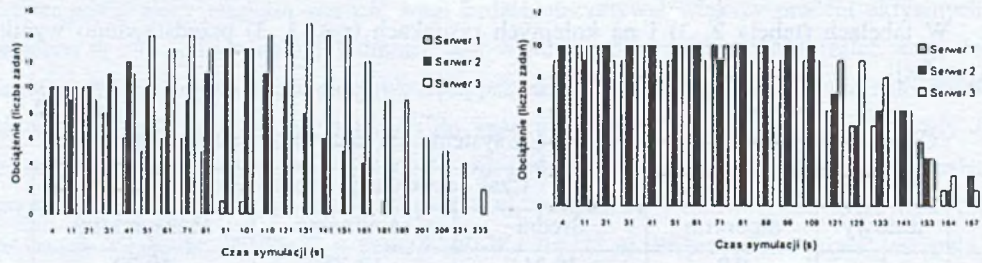
a) b)



Rys. 2. Wyniki symulacji dla pierwszego scenariusza testowego dla algorytmów RR (a) i NLP (b)

Fig. 2. Simulation results for the first testing scenario for algorithms RR (a) and NLP (b)

a) b)



Rys. 3. Wynik symulacji dla drugiego scenariusza testowego dla algorytmów RR (a) i NLP (b)

Fig. 3. Simulation results for the second testing scenario for algorithms RR (a) and NLP (b)

Dla pierwszego scenariusza testowego, gdzie parametry serwerów w klasterze były identyczne, rozdział obciążenia jest raczej akceptowalny (rys. 2) i nie zależy wiele od typu wykorzystywanego algorytmu rozdziału żądań. Sprawa się znacznie komplikuje, kiedy wydajność poszczególnych serwerów wewnątrz klastera jest różna, co ilustruje drugi scenariusz testowy (rys. 3). Wykresy pokazują, że algorytm RR nie radzi sobie dobrze z

równomiernym rozdziałem zapytań pomiędzy serwery, powodując przeciążenie niektórych z nich. Dalsze przydzielanie zapytań do przeciążonych serwerów powoduje dalszy wzrost ich obciążenia, co nie jest korzystne. Algorytm NLP dzięki ciągłemu sprawdzaniu liczby aktywnych połączeń do poszczególnych serwerów przydziela zapytania bardziej równomiernie, nie powodując przeciążenia serwerów. Co więcej, algorytm ten jest o około 17% wydajniejszy, ale tylko w przypadku niejednolitych parametrów klastera. W celu zwiększenia wydajności klastera można zwiększać wydajność poszczególnych serwerów lub też dodawać kolejne serwery do klastera. W rozpatrywanym przypadku dodając kolejny serwer do klastera wydajność zwiększyła się o około 27%.

5. Podsumowanie

Rozwój Internetu wymaga odpowiednich narzędzi do zarządzania klasterem serwerów WWW. Wybór praktycznego rozwiązania dla konkretnych warunków jest raczej trudnym zadaniem. Rozwiązania sprzętowe, takie jak CISCO LocalDirector, na pewno są rozwiązaniami efektywniejszymi, ale znacznie kosztowniejszymi. Alternatywą może być programowe narzędzie, jakim jest IBM SecureWay Network Dispatcher, które oferuje nieznacznie mniejszą wydajność przy znacznym obniżeniu kosztów. Ostatnią już drogą wyboru jest zastosowanie darmowego narzędzia, jakim jest Linux Virtual Serwer, które oferuje podobne możliwości w porównaniu z produktami komercyjnymi, przy znacznie niższych kosztach. Dlatego też w pracy skupiono się na rozwiązaniu LVS.

W pracy porównano dwa reprezentatywne algorytmy alokacji. Algorytm RR należy do algorytmów statycznych, które nie potrzebują żadnej informacji na temat parametrów klastera. Algorytm NLP należy do dynamicznych algorytmów, które w sposób ciągły wykorzystują informację na temat parametrów klastera. Wyniki eksperymentalne pokazują, że jeśli mamy do czynienia z klasterem, w którym parametry poszczególnych serwerów są takie same, to zarówno algorytm statyczny, jak i dynamiczny dają podobne wyniki. Jeśli jednak zmienia się wydajność serwerów w klasterze, co w praktyce ma częściej miejsce. algorytm NLP znacznie skuteczniej równoważy obciążenie, oferując przy tym większą wydajność systemu.

W celu zwiększenia wydajności serwisu WWW można zwiększyć wydajność poszczególnych serwerów w klasterze poprzez dobór odpowiedniego algorytmu alokacji lub zwiększenie liczby serwerów. Prezentowany symulator może być wykorzystywany do wyboru odpowiedniej techniki zrównoważenia obciążenia klasterów WWW i wyznaczenia jego optymalnych parametrów.

LITERATURA

1. Cardellini V., Colajanni M., Yu P.: Dynamic Load Balancing on Web-Server Systems. IEEE Internet Computing, 1999, s. 28-29.
2. Bourke T.: Wyrównywanie obciążenia serwerów. RM, Warszawa 2002.
3. Linux Virual Server Project: http://www.linuxvirtualserver.org.
4. Zhang W.: Linux Virtual Server for Scalable Network Services. http://www.linuxvirtualserver.org/ols/lvs.ps.gz.
5. Filak P.: Porównanie i analiza strategii wykorzystania zasobów sieci WWW. Praca magisterska Wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej wykonana pod kierunkiem H. Krawczyka, Gdańsk 2001.
6. Cisco LocalDirector: http://www.cisco.com/warp/public/cc/pd/cxsr/.
7. IBM SecurWay Network Dispatcher: http://www.ibm.com/software/network/dispatcher/

Recenzent: Prof. dr hab. inż. Andrzej Grzywak

Wpłynęło do Redakcji 9 września 2002 r.

Abstract

The paper shows haw to use a computer cluster to increase efficiency of Web servers. Web cluster server consist of a one management server and many computing servers connected by LAN. For such architecture different server cooperation techniques are proposed and discussed (Fig. 1). They based on the following approaches: Network Address Translation (NAT), Tunneling IP (TUN), Direct Routing (DR). It is shown that technique NAT is less efficient, but easy to implement for small number of servers (about 10). However techniques TUN and DR can be used even for 100-node cluster.

The second problem considered in the paper is evaluation of user request allocation on nodes of the cluster. We consider the following algorithms: round-robin (RR), weighted round-robin (WRR), the smallest number of connections (SNC), weighted SNC (WSNC).

The first algorithms represent static allocation, the next two ones – dynamic allocation. The pseudocodes of the above algorithms are given and their basic characteristics and criteria of comparison are discussed. To evaluate their suitability for Web server cluster the simulator is proposed. To illustrate simulator functionality three simple testing scenarios are defined, realized and the obtained results are shown in Fig. 2 and Fig. 3. For the homogenous cluster both algorithms RR and SNC leads to more acceptable load balancing in comparison to the

heterogeneous cluster. Besides Table 2 and Table 3 shown the response time of the web-server cluster as a function of the number of the used allocation algorithms, and the number of nodes in the cluster, respectively. It was shown, that algorithm SNC is much more useful then algorithm RR.

Adresy

Arkadiusz URBANIAK: Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, ul. Gabriela Narutowicza 11/12, 80-952 Gdańsk Wrzeszcz, Polska, arkur@eti.pg.gda.pl .

Henryk KRAWCZYK: Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, ul. Gabriela Narutowicza 11/12, 80-952 Gdańsk Wrzeszcz, Polska, hkrawk@eti.pg.gda.pl .