

Piotr BRUDŁO

Półtechnika Gdańska, Katedra Architektury Systemów Komputerowych

## STRATEGIE RÓWNOWAŻENIA OBCIĄŻENIA W KOMPUTEROWYCH SIECIACH LOKALNYCH

**Streszczenie.** W artykule sklasyfikowano oraz przeanalizowano najczęściej wykorzystywane strategie równoważenia obciążenia w sieciach lokalnych. Zwrócono uwagę na istotność formalnej definicji obciążenia dla stacji oraz przedstawiono praktyczne aspekty określania zdefiniowanego obciążenia. Zaakcentowano wpływ dokładności oraz aktualności informacji o obciążeniu na efektywność algorytmów dystrybucji zadań. Przedstawiono otwartą strukturę systemu monitorującego obciążenie w sieciach lokalnych, a następnie zaprezentowano implementacje takich systemów dla sieci lokalnych pracujących pod systemami MS Windows oraz Unix.

## LOAD BALANCING STRATEGIES FOR LOCAL AREA COMPUTER NETWORKS

**Summary.** In the article, load balancing strategies applicable for local area networks are classified and analysed. Significance of formal definition of workload for an individual station is spotlighted, and practical aspects of determination of the workload are discussed. Influence of actuality and accuracy of workload information onto efficiency of task distribution algorithms is emphasised. Open software structure of a load monitoring package is shown, and implementations of two monitoring packages for local area networks running MS Windows and Unix are presented.

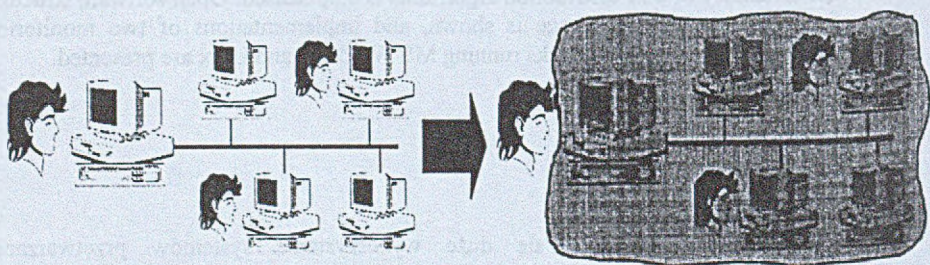
### 1. Wprowadzenie

W chwili obecnej zauważa się duże wykorzystanie systemów przetwarzania rozproszonego i równoległego [1, 2]. Systemy tego typu są atrakcyjne zarówno ze względu na możliwe do uzyskania wielkości mocy obliczeniowych, jak i na możliwości równoległej realizacji określonych funkcji czy operacji. Aktualnie w środowiskach komputerowych

można zauważyć dwie tendencje. Z jednej strony instalowane są duże komputery (ang. mainframes) służące do specjalizowanych zadań typu symulacje skomplikowanych procesów fizykochemicznych, modelowania pogody i atmosfery itp., gdzie wymagana jest koncentracja dużej mocy obliczeniowej i zaangażowanie ich w całości do wykonania określonych zadań. Komputery mainframe instalowane są zwykle w dużych organizacjach ze względu na bardzo wysokie koszty zarówno sprzętu komputerowego oraz równie wysokie (lub niekiedy wyższe) koszty specjalizowanego i zaawansowanego oprogramowania. Ze strony drugiej, dla typowych zastosowań w pełni wystarczające są stacje robocze typu PC czy porównywalne, które łączone są w sieci lokalne (ang. local area networks - LANs) w celach współdzielenia zasobów typu dyski, drukarki czy plotery, a także integracji z zasobami światowej sieci Internet [1, 4].

W sieciach lokalnych sumaryczna moc obliczeniowa stacji roboczych może być porównywana z komputerami typu mainframes, a szybkie i niezawodne połączenia w ramach sieci LAN pozwalają na integrację zasobów obliczeniowych oraz łączne ich wykorzystanie do realizacji określonych zadań. Obecnie proponowanych jest wiele środowisk umożliwiających integrację logiczną stacji roboczych sieci komputerowych [1, 2]. Na rys. 1 pokazano ideę integracji zasobów obliczeniowych sieci LAN w wirtualną maszynę obliczeniową. Wykorzystanie sieci LAN jako zintegrowanego logicznie zasobu obliczeniowego wydaje się być bardzo atrakcyjne ze względu na możliwości uzyskania dużej mocy przetwarzania w oparciu o istniejącą już bazę sprzętową przy jednoczesnym zachowaniu integralnych, a zarazem pożądaných cech sieci komputerowych, takich jak: rozszerzalność, otwartość, elastyczność, niezawodność, przezroczystość etc. [1, 4].

Problemem do rozwiązania pozostaje jednak sposób logicznej integracji fizycznie rozproszonych zasobów obliczeniowych sieci LAN w oparciu o istniejące możliwości i media komunikacyjne, a także efektywne wykorzystanie tych zasobów w celu realizacji określonego typu przetwarzania zadań.



Rys. 1. Logiczna integracja zasobów lokalnej sieci komputerowej w maszynę wirtualną  
Fig. 1. Logical integration of resources of a local area network into a virtual machine



## 2. Efektywne przetwarzanie rozproszone

Integracja logiczna zasobów obliczeniowych sieci lokalnej pozwala na uzyskanie dużych mocy obliczeniowych. W praktyce część zadań kierowana jest na stacje mniej obciążone, uzyskując w efekcie szybsze wykonanie tych zadań. Podstawowym problemem do rozwiązania pozostaje zaproponowanie praktycznej definicji miary obciążenia dla pojedynczej stacji roboczej. Miara ta wykorzystywana będzie do porównań wielkości obciążenia i dlatego w praktyce powinna mieć ona charakter numeryczny lub przynajmniej operować w zakresie dających się uszeregować wielkości nienumerycznych.

W literaturze proponowanych jest wiele różnych miar obciążenia [7, 8, 9, 10, 11, 13]. Można tutaj wymienić: ilość procesów, ilość aktywnych procesów, zagregowana ilość aktywnych procesów, czas zajętości procesora, ważony czas zajętości procesora z uwzględnieniem zajętości pamięci operacyjnej, złożony wskaźnik obciążenia oraz wiele innych. Zastosowanie określonej miary obciążenia dla opisu stanu stacji roboczej zależy zarówno od charakteru przetwarzania, jak i charakteru stacji roboczych. Jak dotychczas, nie wypracowano obiektywnej miary obciążenia dającej się zastosować w ogólnym przypadku.

Przy przyjmowaniu definicji obciążenia stacji roboczej należy mieć na uwadze możliwość określenia aktualnego obciążenia, ale również także możliwość określania obciążenia po uruchomieniu danego zbioru zadań lub pojedynczego zadania na stacji. Precyzyjniej, oznacza to określenie obciążenia wnoszonego przez pojedyncze zadanie wynikające z wartości parametrów tego zadania. Wśród rozważanych parametrów zadania dla przykładu można podać standardowy czas jego wykonania, priorytet, wymagania dotyczące zajętości pamięci i urządzeń, ilość dostępow do dysków etc. Określanie obciążeń wprowadzanych na stacje robocze przez poszczególne zadania konieczne jest w procesie efektywnej dystrybucji zadań.

Miara obciążenia zdefiniowana formalnie musi być efektywnie możliwa do określenia bądź też efektywnie mierzalna. W praktyce oznacza to, że wielkość obciążenia może być albo obliczona na podstawie określonych parametrów stanu stacji roboczych (takich jak np. procent zajętości procesora, stopień zajętości pamięci etc.), albo pomierzona na podstawie czasów wykonania zadań testowych (ang. benchmarków). Obliczanie obciążenia stacji roboczej na podstawie parametrów jej stanu odbywa się bezpośrednio w oparciu o definicję obciążenia (wzór algebraiczny). Przy wykorzystaniu zadania testowego obciążenie określane jest poprzez zachowanie tego zadania (zwykle mierzone jest opóźnienie przy wykonaniu benchmarku). Stąd wymagane jest, aby zadanie testowe było reprezentatywne, tj. wywoływało wpływ na wszystkie parametry zawarte w definicji obciążenia.

Zwykle za miarę efektywności przetwarzania w sieci przyjmuje się zrównoważenie obciążenia dla wszystkich stacji [9, 14]. W praktyce stan idealnego zrównoważenia

obciążenia jest bardzo trudny do osiągnięcia i stąd potrzeba definicji wielkości wyrażającej globalnie stan nierównoważenia. Wielkość ta jest przedmiotem minimalizacji w procesie dystrybucji zadań. Przyjmuje się tutaj np. sumę bezwzględnych wartości odchyień w stosunku do stanu idealnego zrównoważenia lub sumę kwadratów tych odchyień. Przyjęcie definicji wielkości nierównoważenia ma wpływ na zachowanie algorytmów dystrybucji zadań.

## 2.1. Przykład modelu przetwarzania

Sieć komputerowa może być opisana poprzez zbiór  $m$  stacji roboczych  $P = \{P_1, P_2, \dots, P_m\}$ . Stacja robocza  $P_i$  ( $i=1..m$ ) ze zbioru  $P$  opisywana jest poprzez stały parametr  $b_i$  ( $i=1..m$ ), reprezentujący względną szybkość przetwarzania dla stacji  $P_i$ . Parametr ten wprowadzony został w celu uwzględnienia ewentualnej heterogeniczności. Zakłada się, że stacje  $P_i$  ( $i=1..m$ ) są całkowicie niezależne [5, 6].

Stacja  $P_i$  ( $i=1..m$ ) otrzymuje niezależne i niezwiązane ze sobą zadania do wykonania opisane zbiorem  $T_i = \{T_{i,1}, T_{i,2}, \dots, T_{i,n_i}\}$ . Zadanie  $T_{i,j}$  ( $j=1..n_i$ ) jest charakteryzowane przez parę:

$$T_{i,j} = (p_{i,j}, w_{i,j}) \quad j=1..n_i \quad (1)$$

gdzie:  $p_{i,j}$  oznacza znormalizowany czas przetwarzania zadania  $T_{i,j}$ , który oznacza czas wykonania tego zadania na nieobciążonej stacji  $P_i$ , dla której  $b_i = 1$ . Parametr  $w_{i,j}$  określa priorytet zadania  $T_{i,j}$  przy założeniu przetwarzania z podziałem czasu przez stacje  $P_i$  ( $i=1..m$ ).

Stacja  $P_i$  ( $i=1..m$ ) może przetwarzać zadania  $T_{i,j}$  ( $j=1..n_i$ ) lokalnie albo inicjować zdalne wykonania poszczególnych zadań  $T_{i,j}$ . Każde z zadań  $T_{i,j}$  ( $i=1..m, j=1..n_i$ ) może być potencjalnie wykonane na każdej stacji roboczej  $P_i$  ( $i=1..m$ ).

Miara obciążenia  $Q_i$  dla stacji  $P_i$  ( $i=1..m$ ) definiowana jest jako suma priorytetów zadań wykonywanych na stacji  $P_i$ . Formalnie, obciążenie  $Q_i$  zdefiniowane jest w (2). Liczbę zadań wykonywanych na  $P_i$  oznaczono jako  $k_i$ .

$$Q_i = \sum_{j=1}^{k_i} w_{i,j} \quad i=1..m \quad (2)$$

W sieciach heterogenicznych wielkość obciążenia  $Q_i$  ( $i=1..m$ ) powinna być znormalizowana z uwzględnieniem  $b_i$ . Znormalizowane obciążenie  $L_i$  wyrażone jest w (3).

$$L_i = \frac{Q_i}{b_i} \quad i=1..m \quad (3)$$

W przypadku zrównoważenia obciążenia wielkość  $L_i$  powinna być jednakowa dla wszystkich stacji roboczych ze zbioru  $P$ . Jako miarę nierównoważenia przyjęto sumę kwadratów odchyień od stanu zrównoważenia idealnego (4).



$$B = \sum_{i=1}^m (L_i - M)^2 \quad (4)$$

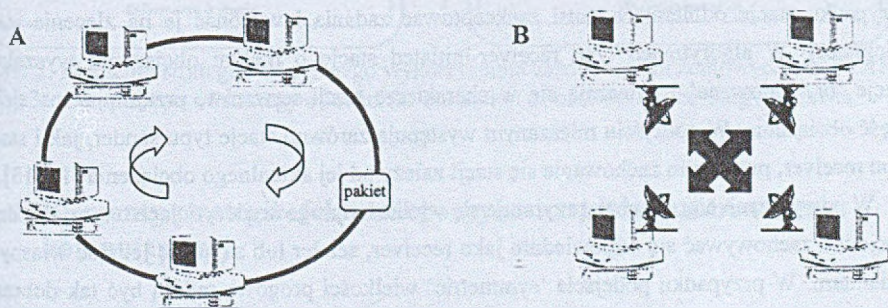
Wielkość  $M$  w wyrażeniu (5) reprezentuje teoretyczną wartość znormalizowanego obciążenia  $L_i$  dla stacji  $P_i$  ( $i=1..m$ ) w stanie idealnego zrównoważenia obciążenia w sieci.

$$M = \frac{\sum_{i=1}^m Q_i}{\sum_{i=1}^m b_i} \quad (5)$$

## 2.2. Wymiana informacji o obciążeniu

Stacja robocza dokonuje dystrybucji zadań, mając na celu osiągnięcie stanu zrównoważenia obciążenia, na podstawie posiadanych informacji o aktualnych obciążeniach pozostałych stacji w sieci. Każda ze stacji dokonuje określenia własnego obciążenia lokalnie, a następnie informuje pozostałe stacje o własnym stanie. Dla wymiany informacji wymagany jest określony schemat komunikacyjny. Najczęściej wykorzystywanymi schematami są komunikacja pierścieniowa oraz komunikacja rozgłoszeniowa. W schemacie komunikacji pierścieniowej stacje połączone są w logiczny pierścień, w którym krąży pakiet zawierający wymagane informacje. W schemacie komunikacji rozgłoszeniowej każda ze stacji nadaje w trybie rozgłoszeniowym (ang. broadcast) informacje o własnym stanie [6].

Wybór ewentualnego schematu powinien być dokonywany w oparciu o takie parametry, jak: narzut komunikacyjny, obciążenie stacji roboczej związane z obsługą danego schematu, opóźnienia komunikacyjne, niezawodność etc. Ponadto należy uwzględnić możliwości rekonfiguracji danego schematu przy ewentualnym dodawaniu bądź odłączaniu stacji w sieci, a także możliwości automatycznych rekonfiguracji w przypadkach awarii. Schematy komunikacji pierścieniowej oraz rozgłoszeniowej zostały pokazane na rys. 2.



Rys. 2. Schematy komunikacji pierścieniowej (A) i rozgłoszeniowej (B)  
Fig. 2. Ring (A) and broadcast (B) communication schemes



### 3. Algorytmny równoważenia obciążenia w sieciach lokalnych

W sytuacjach gdy strumienie zadań do wykonania dla poszczególnych stacji są różne, a zadania wykonywane są jedynie lokalnie, obserwowana jest nieefektywność wykorzystania sieci jako całości. Część stacji jest 'przeładowana', podczas gdy w sieci występują stacje słabo obciążone lub wręcz bezczynne. Pojawia się więc konieczność przekazania części zadań w celu poprawy efektywności wykorzystania sieci jako całości, przy czym przez kryterium efektywności rozumiane jest zrównoważenie obciążenia [7, 9, 10].

Wyróżnia się dwa typy algorytmów równoważenia obciążenia. Pierwszy typ to algorytmy statyczne (ang. static). W algorytmach statycznych decyzja o dystrybucji zadań podejmowana jest jedynie w oparciu o parametry zadań bez dynamicznego sprawdzania informacji o obciążeniach stacji zdalnych. W tego typu algorytmach zakłada się określone probabilistycznie zachowanie obciążenia stacji roboczych. Ewidentną wadą jest tutaj nieprzydatność tych algorytmów dla dynamicznych środowisk, dla których opis probabilistyczny jest bardzo trudny do analitycznego zdefiniowania. Zaletą jest natomiast prostota. Tego typu podejście stosuje się czasem przy organizacji maszyny wirtualnej całkowicie wydzielonej z otoczenia, gdzie zakłada się, że stan obciążenia stacji roboczych wynika tylko z zadań dystrybuowanych w ramach danego systemu [12, 13].

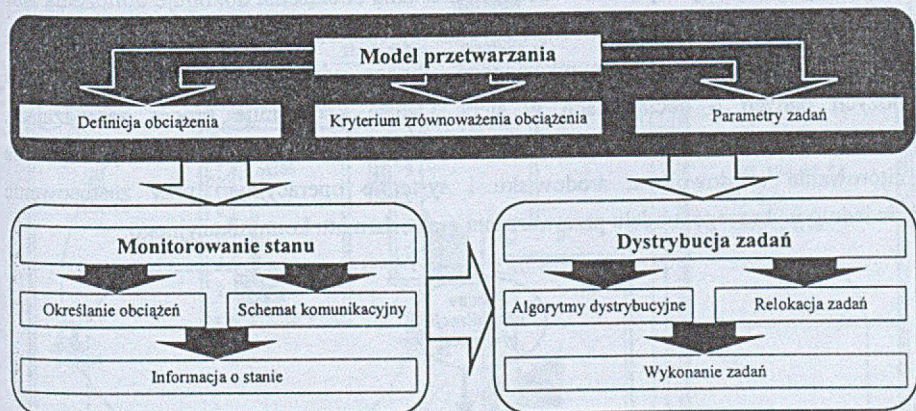
W algorytmach dynamicznych decyzje o dystrybucji zadań podejmowane są na podstawie aktualnych informacji o stanach obciążeń stacji. Stąd podstawą jest tutaj dynamiczne monitorowanie. Generalnie dynamiczne algorytmy równoważenia obciążenia można podzielić na 3 klasy: inicjowane przez stację wysyłającą (ang. sender-initiated), inicjowane przez stację odbierającą (ang. receiver-initiated) oraz klasę będącą połączeniem dwóch poprzednich klas (ang. symmetric). W algorytmach typu sender-initiated stacja, która otrzymała zbiór zadań do przetworzenia, wyszukuje na podstawie znanych informacji o obciążeniach w sieci stacje mało obciążone i do tych stacji dystrybuuje zadania. W tym przypadku stacja odbierająca musi zaakceptować zadania i wykonać je na zlecenie stacji inicjującej. W algorytmach typu receiver-initiated stacje o małym obciążeniu wyszukują stacje 'przeładowane' i zgłaszają się w charakterze stacji-serwerów, przejmując na siebie część obciążenia. W podejściu mieszanym występują zarówno stacje typu sender, jak i stacje typu receiver, przy czym zachowanie się stacji zależy od jej aktualnego obciążenia [12, 13].

W powyższych algorytmach przyjmuje się wielkości progowe decydujące o tym, czy dana stacja ma zachowywać się odpowiednio jako receiver, sender lub zająć się jedynie własnymi zadaniami. W przypadku podejścia 'symmetric' wielkości progowe muszą być tak dobrane, aby nie powodować efektu migotania zadań polegającego na zbyt częstych zmianach charakteru danej stacji i wielokrotnym przesyłaniu tych samych zadań pomiędzy stacjami.



Miara efektywności wykorzystania zasobów obliczeniowych wyrażona jest analitycznie jako stopień zrównoważenia obciążenia. Z punktu widzenia indywidualnej stacji roboczej proces dystrybucji zadań ma charakter rozdziału zadań o określonych parametrach na zbiór stacji w oparciu o miary obciążeń tych stacji i generalnie może być teoretycznie sprowadzony do określonych przypadków problemów typu plecakowego. Możliwe do zastosowania tutaj algorytmy mają złożoność przynajmniej NP-zupełną w zakresie problemów decyzyjnych oraz NP-trudną w zakresie problemów optymalizacyjnych, stąd konieczność stosowania podejść heurystycznych w praktyce [3].

Algorytmy heurystyczne nie dają rozwiązań optymalnych, tym niemniej rezultaty uzyskiwane przy ich wykorzystaniu pozwalają w pełni na ich zastosowanie praktyczne [3, 11]. W przypadku problemu dystrybucji zadań w celu zrównoważenia obciążenia w sieci algorytmy dystrybucyjne opierają się na informacji o obciążeniach w sieci. Należy tutaj podkreślić, że od dokładności oraz aktualności tej informacji zależy uzyskanie pożądanego efektu. W zastosowaniach praktycznych zwraca się znacznie większą uwagę na zapewnienie dokładnej i aktualnej informacji o obciążeniach niż na nieefektywności algorytmów heurystycznych (których i tak nie można już wiele ulepszyć).



Rys. 3. Elementy strategii efektywnego wykorzystania zasobów obliczeniowych sieci LAN  
 Fig. 3. Components of strategy of efficient utilisation of network processing resources

#### 4. Elementy strategii efektywnego wykorzystania zasobów sieci komputerowej

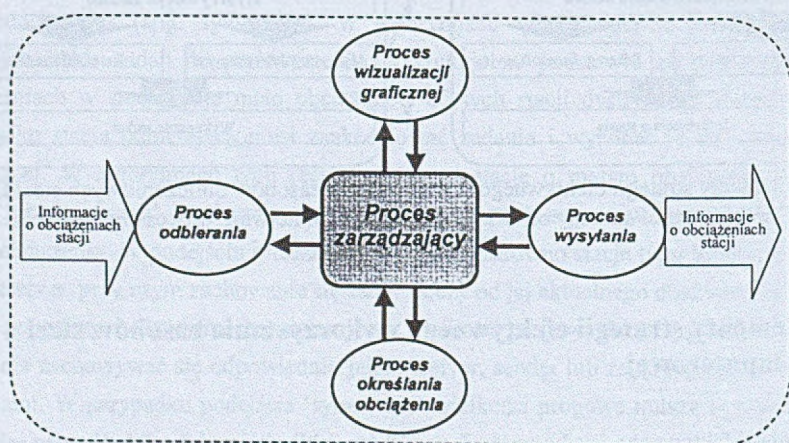
Strategia efektywnego wykorzystania zasobów obliczeniowych składa się z wielu elementów. Na rys. 3 pokazano wzajemne powiązania tych elementów. Podstawowym elementem składowym jest model przetwarzania. Z modelu wynika bezpośrednio analityczna



miara obciążenia stacji roboczej oraz definiowane są parametry przetwarzanych zadań. Z natury definicji obciążenia wynika sposób jego określania: obliczanie na podstawie parametrów stanu stacji roboczej lub poprzez pomierzenie. Kolejnym elementem jest schemat komunikacyjny umożliwiający wymianę informacji o obciążeniach pomiędzy stacjami zapewniający aktualność tej informacji. Określanie stanu obciążeń wraz ze schematem komunikacyjnym składają się na monitorowanie stanu. Na podstawie dokładnej i aktualnej informacji o stanie dokonywana jest dystrybucja zadań w celu zrównoważenia obciążenia oraz następnie wykonanie zadań.

## 5. System monitorowania obciążenia

Ze względu na istotność dokładności oraz aktualności informacji o obciążeniach stacji roboczych szczegółowo opracowano strukturę systemu monitorującego obciążenie w sieci lokalnej i następnie zaimplementowano go w środowisku Unix oraz MS Windows. Struktura oprogramowania systemu monitorowania pokazana została na rys. 4. Oprogramowanie składa się z pięciu podstawowych procesów. Proces określania obciążenia dokonuje obliczenia albo pomierzenia wielkości lokalnego obciążenia. Procesy odbierania i wysyłania odpowiedzialne są za realizację schematu komunikacyjnego. Proces wizualizacji graficznej dokonuje ekspozycji danych o obciążeniach w sieci. Całość koordynuje proces zarządzający. Zaproponowana struktura może zostać wykorzystana do implementacji systemu monitorowania w dowolnym środowisku i systemie operacyjnym przy zastosowaniu odpowiednio wybranego modelu przetwarzania oraz schematu komunikacyjnego.



Rys. 4. Struktura systemu monitorowania obciążenia w sieci lokalnej

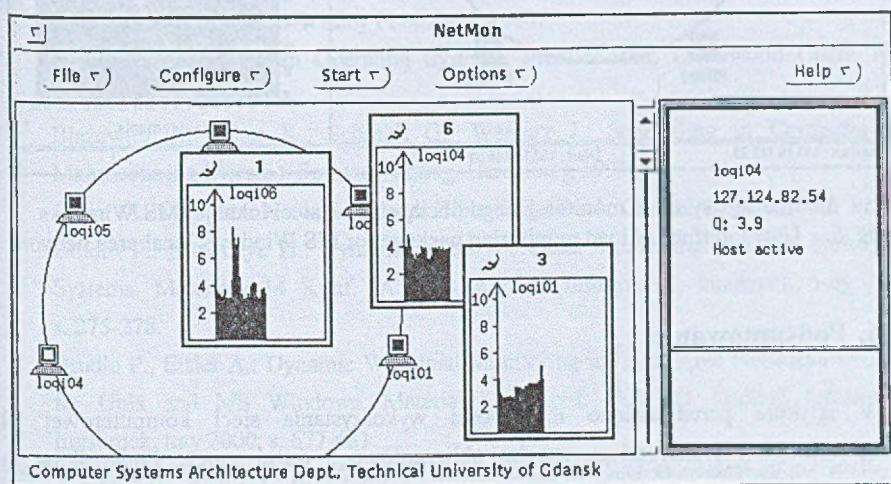
Fig. 4. Structure of load monitoring system oriented for local area network



### 5.1. Implementacja systemu monitorowania obciążeń stacji roboczych

W oparciu o strukturę oprogramowania zaproponowaną na rys. 4 zaimplementowano systemy monitorujące obciążenie dla lokalnych sieci komputerowych pracujących odpowiednio pod systemami operacyjnymi MS Windows oraz Unix. Zastosowano pierścieniowy schemat komunikacji. Na rys. 5 pokazano interfejs użytkownika dla systemu monitorującego obciążenie w sieci pracującej systemem SunOS ver. 5 w środowisku graficznym OpenWindows lub Solaris CDE. Operacje komunikacyjne zaimplementowano w języku C/C++ przy wykorzystaniu połączeniowych gniazdek protokołu TCP/IP. Interfejs użytkownika zgodny jest ze standardem XView/OpenLook.

W oknie głównym przedstawiony jest logiczny pierścień ze stacjami roboczymi. Dla każdej stacji otwierane jest okno, w którym prezentowany jest histogram obciążenia. Ponadto system posiada wiele funkcji dotyczących organizacji pierścienia logicznego typu dodawanie i odłączanie stacji oraz inne funkcje dla organizacji konfiguracji. Dane o obciążeniach stacji roboczych przechowywane są we współdzielonej tablicy i mogą być w łatwy sposób wykorzystane przez inne aplikacje (np. systemy dystrybucji zadań).



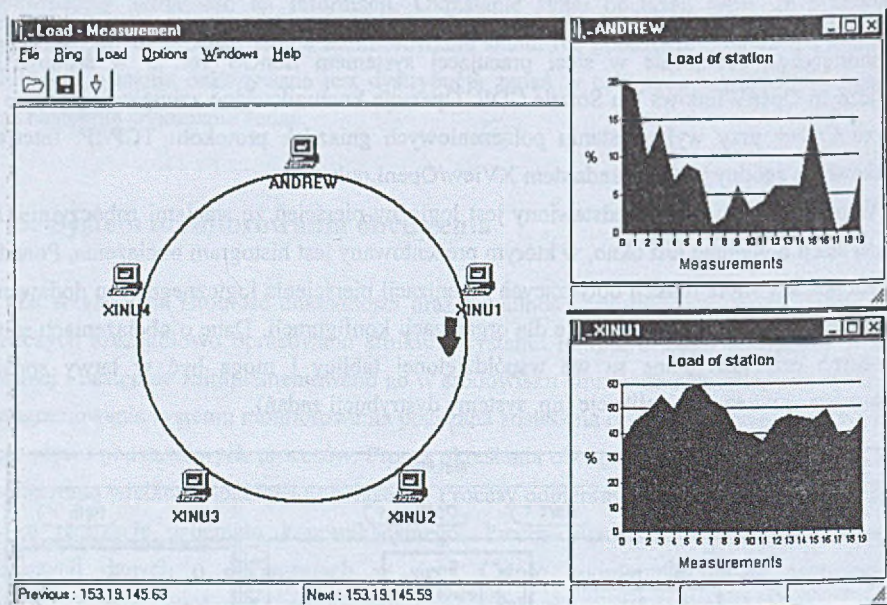
Rys. 5. Interfejs systemu monitorującego obciążenie w sieci lokalnej Unix

Fig. 5. User interface of load monitoring package for Unix local area network

Interfejs użytkownika systemu monitorującego obciążenie dla systemu operacyjnego MS Windows przedstawiono na rys. 6. Założenia projektowe dla oprogramowania są zgodne ze strukturą pokazaną na rys. 4. System zaimplementowany został w języku C/C++ przy wykorzystaniu środowiska Borland C++ Builder. Komunikacja odbywa się poprzez gniazdko protokołu TCP/IP. System został uruchomiony i testowany w systemach Windows 95/98/NT.



W głównym oknie prezentowany jest logiczny pierścień. Dla każdej ze stacji możliwe jest otwarcie okna z przebiegiem funkcji obciążenia. Ponadto możliwe jest również wyświetlanie histogramów zbiorczych oraz wybranych statystyk. System oferuje także funkcje dotyczące rekonfiguracji pierścienia logicznego oraz wiele innych.



Rys. 6. Interfejs systemu monitorującego obciążenie w sieci lokalnej MS Windows  
Fig. 6. User interface of load monitoring package for MS Windows local area network

## 6. Podsumowanie

W artykule przedstawiono możliwości wykorzystania sieci komputerowej jako zintegrowanego i jednorodnego zasobu obliczeniowego dużej mocy. Podkreślono rolę przyjętego modelu przetwarzania. Pokazano, że z modelu przetwarzania wynikają bezpośrednio definicja obciążenia oraz definicja kryterium efektywnego wykorzystania zasobów sieci komputerowej. Jako kryterium efektywności przyjęto stopień zrównoważenia obciążenia. Przedstawiono przykład takiego modelu przy założeniu przetwarzania zadań z podziałem czasu na indywidualnych stacjach. Zaprezentowano i przeanalizowano podstawowe typy algorytmów dystrybucji zadań. Zwrócono uwagę na złożoność obliczeniową tych algorytmów. Podkreślono istotność dokładności oraz aktualności informacji o obciążeniach w sieci w procesie efektywnej dystrybucji zadań dokonywanym



przez indywidualne stacje robocze. Skoncentrowano się na systemie monitorowania obciążeń stacji w sieci lokalnej. Pokazano, że w systemie takim podstawową rolę pełni proces dynamicznego określania obciążenia oraz schemat komunikacyjny służący wymianie informacji. Zaproponowano otwartą strukturę systemu monitorującego obciążenie w sieciach lokalnych. Zaimplementowano i zaprezentowano systemy monitorowania pracujące odpowiednio w systemach operacyjnych Unix oraz MS Windows wykorzystujące okienkowe interfejsy graficzne dla prezentacji gromadzonych informacji.

Powyższe systemy monitorowania mogą stanowić podstawę dla systemów dynamicznej dystrybucji zadań. W systemie dystrybucji zadań wyszczególniono funkcje dotyczące przydziału zadań do stacji roboczych, a także funkcje rlokacji zadań oraz zdalnego bądź lokalnego ich wykonania.

## LITERATURA

1. Colouris G., Dollimore J., Kindberg T.: *Distributed Systems. Concepts and Design*. second edition, Addison-Wesley Publishers, 1994.
2. Tanenbaum A.: *Modern Operating Systems*. Prentice-Hall, Englewood Cliffs, New Jersey 1995.
3. Błażewicz J., Ecker K., Schmidt G., Węglarz J.: *Scheduling in Computer and Manufacturing Systems*. Springer-Verlag, Heidelberg 1993.
4. Gibbs M.: *Computer Networks. User's Bible*. Sams Publ., Carmel, Indiana 1993.
5. Brudło P., Krawczyk H.: *Dynamic Node Workload Monitoring in Local Networked Systems*. Materiały 14 Konf. IASTED Applied Informatics, Innsbruck, luty 1996, s. 275-278.
6. Brudło P., Eisler A.: *Dynamic Workload Monitoring in Local Area Networks Oriented for Unix and MS Windows*. Materiały 18 Konf. IASTED Applied Informatics, Innsbruck, luty 2000, s. 677-681.
7. Anthony R., Austin J.: *An Empirical Investigation of Load Sharing Policies*. Materiały 18 Konf. IASTED Applied Informatics, Innsbruck, luty 2000, s. 682-688.
8. Anthony R., Austin J.: *Transparent Automated Load Sharing*. Materiały 18 Konf. IASTED Applied Informatics, Innsbruck, luty 2000, s. 689-694.
9. C. Burdorf, J. Marti: *Load Balancing Strategies for Time Warp on Multi-User Workstations*. *Computer J.*, vol. 36(2), 1993, s. 168-176.
10. Baumgartner K., Wah B.: *GAMMON: A Load Balancing Strategy for Local Computer Systems with Multiaccess Networks*. *IEEE Trans. Comp.*, vol. 38, no. 8, sierpień 1989, s. 1098-1109.



11. Lo V.: Heuristic Algorithm for Task Assignment in Distributed Systems. IEEE Trans. Comp., vol. 37, no. 11, listopad 1988, s. 1384-1397.
12. Shin K., Chang Y.-C.: Load Sharing in Distributed Real-Time Systems with State-Change Broadcasts. IEEE Trans. Comp., vol. 38, no. 8, sierpień 1989, s. 1124-1142.
13. Mirchandaney R., Towsley D., Stankovic J.: Analysis of the Effects of Delays on Load Sharing. IEEE Trans. Comp., vol. 38, no. 11, listopad 1989, s. 1513-1525.
14. Eager D., Zahorjan J., Lazowska E.: Speed-up Versus Efficiency in Parallel Systems. IEEE Trans. Comp., vol. 38, no. 3, marzec 1989, s. 408-423.

Recenzent: Prof. dr hab. inż. Andrzej Grzywak

Wpłynęło do Redakcji 27 marca 2000 r.

### Abstract

In the article, a discussion about computer environments is taken. Two boundary tendencies are presented: on one hand mainframes with their specialised areas of application, and everywhere present local area networks (LANs) with their advantages, on the other. LAN features like openness, expandability and extendibility, flexibility, dependability and transparency are highlighted. A concept of logical integration of LAN processing resources into a virtual machine with processing power comparable to mainframe's is addressed. The concept is illustrated in Fig. 1. Foundations of efficient distributed processing are discussed. Significance of formal definition of workload is spotlighted. Efficiency of utilisation of LAN resources is defined in terms of uniform load distribution. Formal model of network processing is proposed. Computer network is described by set of heterogeneous stations  $P = \{P_1, P_2, \dots, P_m\}$ . Definitions of: load  $Q_i$  and normalised load  $L_i$  for station  $P_i$  ( $i=1..m$ ), network load balance measure  $B$  and theoretical value of normalised load in case of uniform load distribution  $M$  are given in formulas (2-5), respectively. Two communication schemes for load information dissemination are compared: ring with circulating packet and broadcasting (Fig. 2). Three types of balancing algorithms are presented: sender-initiated, receiver-initiated and symmetric. Complexity of load distribution algorithms is classified as non-polynomial and obligation of heuristic approach is discussed. Influence of accuracy and actuality of workload information onto the task distribution algorithms is emphasised. Open software structure of load monitoring package is proposed (Fig. 4). Implementations of two monitoring packages for LANs running MS Windows and Unix are presented (Fig. 5, Fig. 6).