ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1995

Seria: INFORMATYKA z.28 Nr kol. 1270

Wiesław BARCIKOWSKI
Paweł KRYSTEK

PODSYSTEM PRZETWARZANIA ROZPROSZONEGO W SYSTEMIE OPERACYJNYM QNX

Streszczenie. W artykule przedstawiono podsystem umożliwiający realizację przetwarzania rozproszonego w systemie operacyjnym QNX. Pozwala on na zdalną realizację zadań w aktualnie dostępnych węzłach sieci lokalnej. Decyzja o wysłaniu zadania do zdalnej realizacji podejmowana jest w sposób automatyczny (bez udziału użytkownika) w oparciu o bieżące obciążenie węzłów tworzących system rozproszony. Wskaźnik obciążenia uwzględnia moc obliczeniową i stopień załadowania węzła zadaniami. Podsystem ten tworzy środowisko (distributed shell) dające wrażenie pracy na dużym jednolitym komputerze, na który w rzeczywistości składają się niezależne mikrokomputery tworzące sieć lokalną.

DISTRIBUTED PROCESSING SUBSYSTEM IN THE QNX OPERATING SYSTEM

Summary. The paper presents distributed processing subsystem based on QNX operating system. Remote execution of tasks in all achievable nodes of LAN is implemented. The decision of remote execution of particular task is taken automatically (without notification of a user) and bases on current workload of distributed system. Workload index includes the number of tasks and computing power of a node. This subsystem creates an environment (distributed shell) making illusion of work with large single computer which in reality consists of independent microcomputers conncected to LAN.

DAS UNTERSYSTEM DER VERTEILTEN DATENVERARBEITUNG IM QNX-BETRIEBSYSTEM

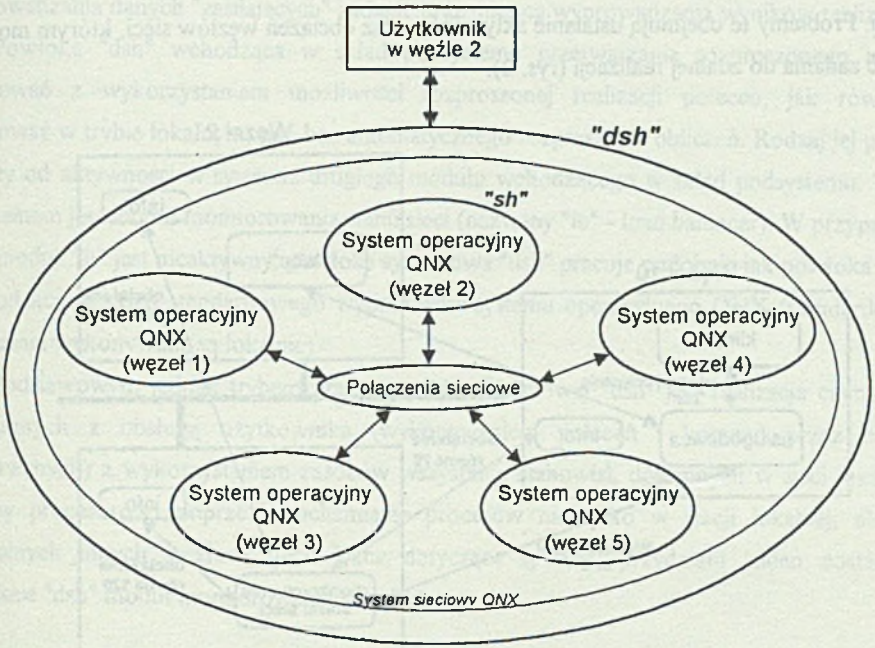
Zusammenfassung. In dem Artikel ist das Untersystem beschrieben, das die verteilte Datenverarbeitung im QNX-Betriebsystem ermöglicht. In diesem System kann

man in allen freien LAN-Knoten fernrechnen. Die Entscheidung über der Fernverarbeitung der Aufgabe ist automatisch (ohne der Benutzer Teilnahme), auf der aktuellen Arbeitslast-Basis in der Knoten, aufgenommen. Der Arbeitslast-Faktor berücksichtig die Rechenleistungund die Zahl der Aufgaben in dem Knoten. Dieses Untersystem ist eine Umwelt, die der Eindruck macht, dass wir mit einem Grossrechner arbeiten, der in Wirklichkeit aus unabhängigen Mikrocomputern der Lokalen-Netz besteht.

1. Przetwarzanie rozproszone

Lata dziewięćdziesiąte przynoszą coraz większe zainteresowanie możliwością łączenia ze sobą odrębnych komputerów. Rozpowszechnienie narzędzi umożliwiających przesyłanie danych pomiędzy nimi okazało się przełomem w rozwoju techniki komputerowej. Pojawiły się sieciowe systemy operacyjne umożliwiające użytkownikowi świadome wykorzystywanie zasobów innych komputerów dołączonych do sieci. Pozwala to spojrzeć na sieć komputerową w inny sposób. Możemy widzieć nie jeden określony komputer, ale cały zespół maszyn z możliwością ich wzajemnej komunikacji. Z takiego punktu widzenia zadanie użytkownika można zlecić do wykonania nie określonej maszynie, ale sieci jako całości. Tego rodzaju działanie przynosi wiele korzyści. Zadania mogłyby być wykonywane w komputerach, które oferują najlepsze warunki do realizacji określonych typów zadań. Poza tym można wykorzystać zasoby (procesory) dostępne w sieci w taki sposób, aby wyrównać ich obciążenie. Można zrealizować to poprzez przydział zadań do określonych węzłów systemu, jak i przekazywanie zadań w trakcie ich wykonywania między komputerami sieci. W przypadku awarii jednego komputera zadania w nim realizowane mogą być przenoszone w inne miejsce. Dane mogłyby być powielone w różnych komputerach, aby w przypadku awarii jednego z nich, zawsze była dostępna inna kopia. Ponadto duże zadania mogłyby być dzielone na mniejsze części (samodzielne procesy), które następnie mogłyby być współbieżnie wykonywane w wielu komputerach. Czas realizacji takich zadań uległby znacznem skróceniu Jak więc widać, systemy rozproszone umożliwiają zrównoleglenie przetwarzania, zapewniają znacznie większą niezawodność oraz bardziej równomierne wykorzystanie zasobów.

Zwykły sieciowy system operacyjny zarządzający zespołem komputerów nie umożliwia realizacji przetwarzania rozproszonego. Dopiero po zaimplementowaniu w nim niektórych mechanizmów można osiągnąć zamierzony cel, jakim może być stworzenie użytkownikowi wrażenia, że korzysta on z jednego dużego systemu (komputera), a nie z sieci złożonej z autonomicznych jednostek (rys. 1).

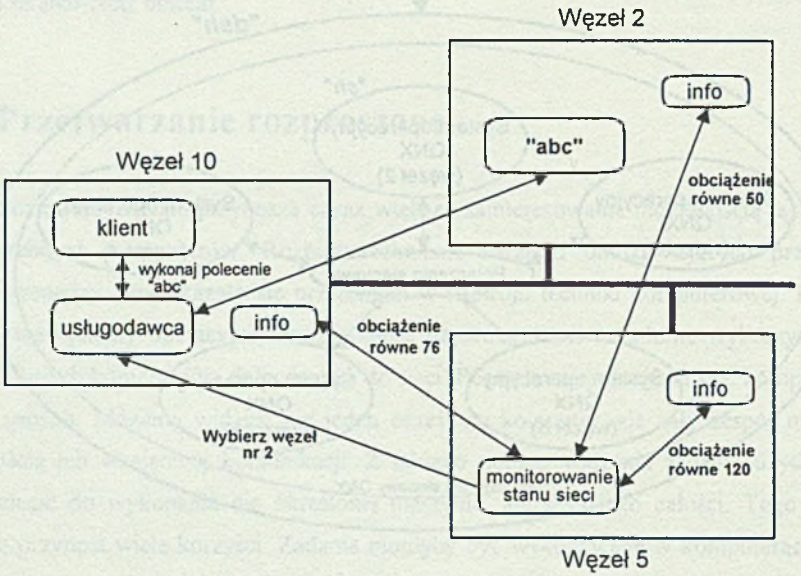


Rys. 1. Struktura systemu rozproszonego opartego o system QNX
Fig. 1. Structure of distributed processing system basing on QNX

2. Idea funkcjonowania podsystemu przetwarzania rozproszonego

Użytkownik pracujący przy jednym z komputerów wchodzących w skład sieci może wykonywać swoje zadania wykorzystując jednostkę, przy której aktualnie pracuje lub też zlecać zadania do wykonania innym jednostkom, specyfikując dokładnie miejsce ich realizacji. W celu realizacji zdalnego wykonywania zadań użytkownik powinien zaopatrzyć się w informację dotyczącą stanu sieci komputerowej (LAN), tzn. aktywności określonych komputerów w danej chwili, a dodatkowo informację o ewentualnej jakości wybranych jednostek (moc obliczeniowa, obciążenie).

Podsystem przetwarzania rozproszonego zrealizowany w systemie operacyjnym QNX pozwala odciążyć użytkownika od problemów związanych z rozproszoną (zdalną) realizacją zadań. Problemy te obejmują ustalanie aktywności oraz obciążeń węzłów sieci, którym można zlecać zadania do zdalnej realizacji (rys. 2).



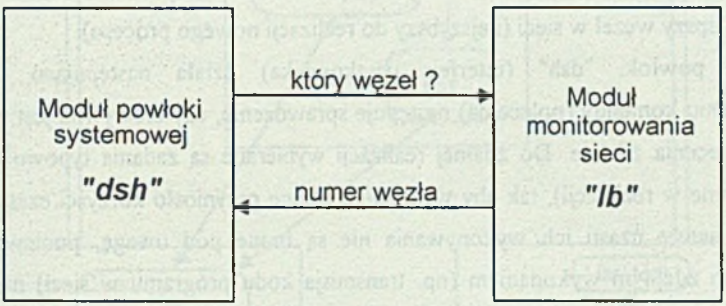
Rys. 2. Schemat działania podsystemu przetwarzania rozproszonego
Fig. 2. Functioning scheme of distributed processing subsystem

Podsystem przetwarzania rozproszonego tworzą dwa moduły. Jednym z nich jest moduł tworzący warstwę izolującą użytkownika od całości sieciowego systemu operacyjnego. Modułem tym jest powłoka systemowa (shell), która jest umieszczona i pracuje w miejsce standardowej powłoki systemowej będącej na wyposażeniu systemu operacyjnego (rys. 1). Nowa powłoka udostępnia identyczne możliwości jak powłoka standardowa, tzn. umożliwia użytkownikowi wprowadzanie i wykonywanie komend (poleceń) systemu operacyjnego. Od strony systemowej powłoka jest procesem działającym w systemie (nazwanym: "dsh" - distributed shell - od charakteru jego pracy umożliwiającego przetwarzanie rozproszone) i realizującym wyżej wymienione funkcje. Zasadniczą różnicą pomiędzy powłokami jest możliwość rozproszonej realizacji zadań przez powłokę "dsh". Standardowa powłoka "sh" umożliwia zdalne wykonywanie zadań pod warunkiem podania przez użytkownika w składni

wprowadzanej komendy numeru węzła, w którym ma być ono wykonywane, a także miejsca wprowadzania danych "zasilających" zadanie oraz miejsca wyprowadzania wyników realizacji.

Powłoka "dsh" wchodząca w skład podsystemu przetwarzania rozproszonego może pracować z wykorzystaniem możliwości rozproszonej realizacji poleceń, jak również pracować w trybie lokalnym, tzn. bez automatycznego rozpraszania obliczeń. Rodzaj jej pracy zależy od aktywności w systemie drugiego modułu wchodzącego w skład podsystemu. Tym elementem jest moduł monitorowania stanu sieci (nazwany "lb" - load balancer). W przypadku gdy moduł "lb" jest nieaktywny, powłoka systemowa "dsh" pracuje podobnie jak powłoka "sh" wchodząca w skład standardowego wyposażenia systemu operacyjnego QNX (standardowo polecenia wykonywane są lokalnie).

Podstawowym jednak trybem pracy powłoki systemowej "dsh" jest realizacja czynności związanych z obsługą użytkownika (wykonywaniem poleceń i komend przez niego wydawanych) z wykorzystaniem zasobów wszystkich stanowisk dostępnych w sieci lokalnej (mocy procesorów) poprzez uruchamianie procesów nie tylko w stacji lokalnej, ale w dowolnych innych węzłach sieci. Dane dotyczące sposobu przydziału zadań dostarcza powłoce "dsh" moduł monitorowania sieci "lb" (rys. 3).



Rys. 3. Komunikacja między modułami podsystemu przetwarzania rozproszonego
Fig. 3. Communication between modules in distributed processing subsystem

Celem działania podsystemu przetwarzania rozproszonego jest efektywne wykorzystanie zasobów przy jednoczesnym skróceniu czasu realizacji poleceń poprzez wykonywanie ich w najlepszych węzłach sieci (najmniej obciążonych, o największej mocy obliczeniowej). Badaniem stanu sieci pod tym kątem zajmuje się moduł "lb", którego zadaniem jest:

- dynamiczna kontrola aktywności poszczególnych węzłów wchodzących w skład sieci lokalnej QNX,
- pomiar obciążenia wszystkich aktywnych węzłów sieci,
- udostępnianie listy numerów węzłów wraz z odpowiadającymi im wskaźnikami obciążenia.

Moduł na bieżąco kontroluje aktywność węzłów sieci. W trakcie sprawdzania wykrywane jest włączanie lub wyłączanie węzła sieci. Oprócz tego moduł przechowuje informacje dotyczące stanu obciążeń poszczególnych węzłów. Dla zamkniętej lokalnej sieci komputerowej można wyznaczyć liczbowy, względny wskaźnik obciążenia stanowiska, którym może być czas wykonania krótkiej procedury testowej. Co pewien okres (ustalony na 2 sekundy) uruchamiany jest w danym węźle sieci krótki proces obliczeniowy. Czas wykonywania tego procesu (mierzony w milisekundach) jest traktowany jako liczbowy wskaźnik zarówno mocy obliczeniowej danej jednostki, jak i jej obciążenia. Wiadomo przecież, że im większa ilość procesów pracuje w węźle, tym dłuższy jest czas wykonywania procedury testowej. Czas ten zależy również od szybkości (mocy obliczeniowej) maszyny. Im szybszy komputer, tym krótszy jest czas realizacji procedury testowej. Zmierzony wskaźnik zostaje przyporządkowany do numeru węzła, któremu odpowiada, i takie dwie wielkości udostępniane są powłoce "dsh". W wyniku porównania wszystkich wskaźników możemy wybrać najlepszy węzeł w sieci (najszybszy do realizacji nowego procesu).

Moduł powłoki "dsh" (interfejs użytkownika) działa następująco. Po każdym wprowadzeniu komendy (polecenia) następuje sprawdzenie, czy efektywne jest wykonywanie danego polecenia zdalnie. Do zdalnej realizacji wybierane są zadania typowo obliczeniowe (czasochłonne w realizacji), tak aby wykonanie zdalne przyniosło korzyści czasowe. Zadania krótkie w sensie czasu ich wykonywania nie są brane pod uwagę, ponieważ czynności związane ze zdalnym wykonaniem (np. transmisja kodu programu w sieci) mogą znacznie wydłużyć ogólny czas zakończenia wykonania takiego zadania. Powłoka "dsh" przydziela zadania do takich węzłów sieci, dla których względny wskaźnik obciążenia jest najmniejszy. Przy wyborze węzła uwzględniana jest różnica pomiędzy wskaźnikiem obciążenia wybranego węzła zdalnego a wskaźnikiem obciążenia węzła lokalnego. W przypadku niewielkiej różnicy obu wskaźników zadanie wykonywane jest lokalnie.

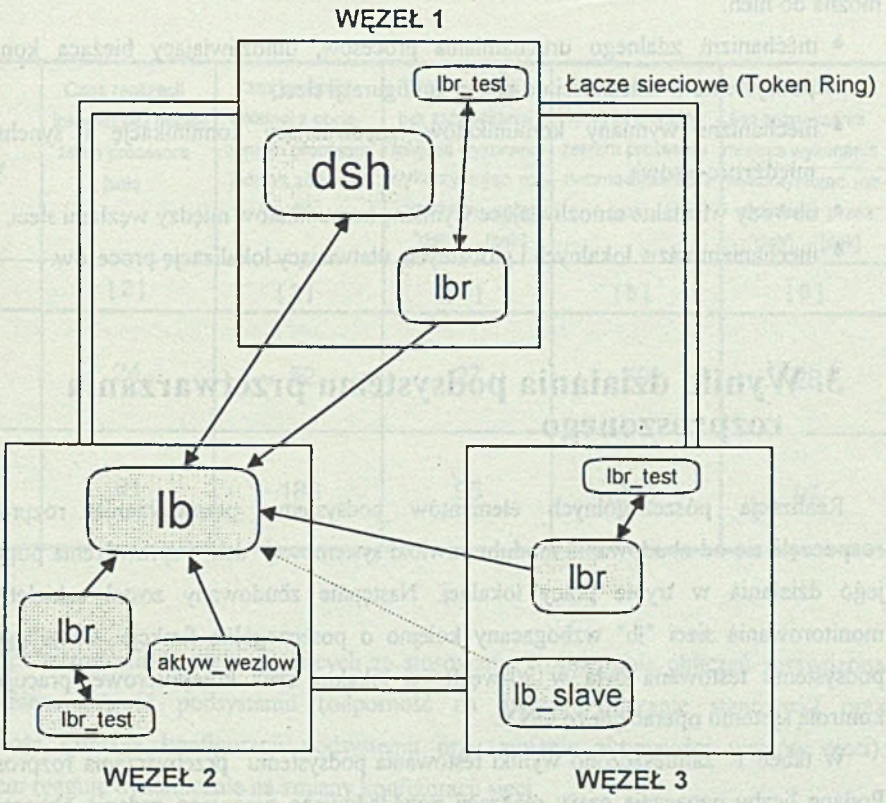
Moduł "lb" współpracujący z modułem "dsh" tworzy w systemie pewna liczba procesów współbieżnych pracujących w różnych węzłach sieci. Nad całością pracy modułu czuwa proces administratora "lb", który to bezpośrednio współpracuje z procesem powłoki systemowej "dsh". Oprócz tego w skład modułu wchodzą następujące procesy (rys. 4):

- proces kontroli aktywności węzłów "aktyw_wezlow"

Dokonuje powyżej opisanej kontroli aktywności węzłów. W przypadku stwierdzenia zmiany konfiguracji sieci proces ten informuje o tym fakcie administratora modułu.

- proces agenta "lbr" wraz z dodatkowym procesem "lbr_test"

Dokonuje pomiaru obciążenia komputera (musi być uruchomiony w każdym aktywnym węźle sieci). Właściwą procedurą testową jest proces "lbr_test" pracujący ze standardowym priorytetem zadań użytkowych. Wydzielenie procesu dokonującego pomiaru obciążenia węzła jest konieczne, ponieważ chroni agenta ("lbr") przed zablokowaniem w przypadku pojawienia się w danym węźle zadania o wysokim priorytecie. Uruchomienie zadania "lbr" bezpośrednio informującego administratora o stanie obciążenia z wysokim priorytetem zabezpiecza przed możliwością zatrzymania przez inne zadanie i umożliwia ciągłą łączność z administratorem. Przekazywanie danych przez agenta odbywa się wówczas, gdy wynik pomiaru uległ zmianie.



Rys. 4. Rozlokowanie procesów w podsystemie przetwarzania rozproszonego

Fig. 4. Location of processes in the subsystem of distributed processing

Na podstawie danych pochodzących z procesu kontrolującego aktywność węzłów administrator odnotowuje każdą zmianę konfiguracji sieci (i ewentualnie uruchamia procesy agentów w nowo dołączonych do sieci węzłach). Dodatkowym elementem, który zwiększa niezawodność pracy modułu monitorowania sieci "lb", jest redundancja procesu administratora (proces administratora jest trzonem podsystemu przetwarzania rozproszonego). Zapasowa kopia procesu administratora (figurująca pod nazwą "lb_slave"), utworzona w innym niż "lb" węźle sieci, pozostaje w stanie gotowości do działania. W przypadku awarii węzła (braku dostępu do administratora) kopia automatycznie przejmuje funkcje administratora "lb". Przełączenie to nie wpływa na ciągłość pracy podsystemu. Wszystkie pozostałe procesy wchodzące w skład podsystemu automatycznie "przełączają się" do nowego administratora.

Konstrukcję podsystemu przetwarzania rozproszonego oraz współpracę poszczególnych procesów wchodzących w jego skład ułatwiają niektóre z mechanizmów s.o. QNX. Zaliczyć można do nich:

- mechanizm zdalnego uruchamiania procesów, umożliwiający bieżącą konfigurację podsystemu w zależności od zmian konfiguracji sieci,
- mechanizm wymiany komunikatów, zapewniający komunikację i synchronizację międzyprocesową,
- obwody wirtualne umożliwiające wymianę komunikatów między węzłami sieci,
- mechanizm nazw lokalnych i globalnych, ułatwiający lokalizację procesów.

3. Wyniki działania podsystemu przetwarzania rozproszonego

Realizacja poszczególnych elementów podsystemu przetwarzania rozproszonego rozpoczęła się od zbudowania modułu powłoki systemowej "dsh" i sprawdzenia poprawności jego działania w trybie pracy lokalnej. Następnie zbudowany został szkielet modułu monitorowania sieci "lb" wzbogacany kolejno o poszczególne funkcje. Całość gotowego podsystemu testowana była w 20-węzłowej lokalnej sieci komputerowej pracującej pod kontrolą systemu operacyjnego QNX.

W tabeli 1 zamieszczono wyniki testowania podsystemu przetwarzania rozproszonego. Podane liczby oznaczają czasy realizacji przykładowego typowego zadania obliczeniowego (zadanie "Wieże Hanoi").

Zadanie to wykonywane było w następujących warunkach:

- przy nieobciążonym procesorze - lokalnie (kolumna [2]),
- przy obciążeniu procesora lokalnego podobnym zadaniem - lokalnie (kol. [3]),
- przy obciążeniu procesora lokalnego podobnym zadaniem - w środowisku rozproszonym "dsh" (kol. [4]),
- przy obciążeniu procesora lokalnego dwoma podobnymi zadaniami - lokalnie (kol. [5]),
- przy obciążeniu procesora lokalnego dwoma podobnymi zadaniami - w środowisku rozproszonym "dsh" (kol. [6]).

Zadanie było testowane dla dwóch parametrów (kol.[1]) określających czas wykonania zadania (ilość krążków do przełożenia między wieżami).

Tabela 1

Wyniki testowania podsystemu przetwarzania rozproszonego

Parametr procedury testowej	Czas realizacji lokalnej bez obciążenia procesora [sek]	Czas realizacji lokalnej z obciążeniem procesora jednym zadaniem [sek]	Czas realizacji bez zaznaczania miejsca wykonania wykorzystując rozproszenie przez "dsh" [sek]	Czas realizacji lokalnej z obciążeniem procesora dwoma zadaniami [sek]	Czas realizacji bez zaznaczania miejsca wykonania wykorzystując rozproszenie przez "dsh" [sek]
[1]	[2]	[3]	[4]	[5]	[6]
1	24	52	27	79	28
2	91	188	95	285	97

Oprócz testowania korzyści płynących ze stosowania rozpraszania obliczeń sprawdzona została niezawodność podsystemu (odporność na awarie-wyłączanie stanowisk) oraz elastyczność (zmiana konfiguracji podsystemu przy zmianie aktywności węzłów sieci). Podsystem reaguje dynamicznie na zmiany konfiguracji sieci.

4. Podsumowanie

Opracowanie to stanowi podstawę do dalszej działalności dotyczącej opracowywania mechanizmów podejmowania decyzji związanych z przydziałem określonych typów zadań do najlepszych dla nich stanowisk (np. czasochłonne obliczenia numeryczne powinny trafić do komputerów wyposażonych w koprocesor matematyczny). Skutkiem rozważań powinno być także rozbudowanie podsystemu przetwarzania rozproszonego o dynamiczną strategię równoważenia obciążenia z możliwością przenoszenia zadań do innych węzłów w trakcie ich realizacji (tzw. migracji procesów). Praca dowiodła, że dysponując bazą sprzętową, jaką jest lokalna sieć komputerowa złożona ze stanowisk klasy IBM PC, oraz systemem operacyjnym QNX, można realizować przetwarzanie rozproszone. Jak się można było spodziewać, tego rodzaju przetwarzanie jest znacznie bardziej efektywne od tradycyjnego. Powinno się dążyć do automatyzacji rozpraszania wszelkich czasochłonnych działań oraz zachęcać do projektowania zadań w taki sposób, aby możliwe było rozdzielenie i zrównoleglenie pracy.

LITERATURA

[1] Barak A., Guday S., Wheeler R.: The MOSIX Distributed Operating System. Springer Verlag, Berlin 1993.

[2] Borzemski L., Głowacz S., Wojtczak K: System równoważenia obciążenia w sieci lokalnej komputerów UNIX. Materiały Seminarium "Sieci komputerowe", Politechnika Śląska, Gliwice 1993.

[3] Coulouris G., Dollimore J.: Distributed Systems - Concepts and Design. Addison-Wesley Publishing Company, 1988.

[4] Weiss Z.: Rozproszone systemy operacyjne. Informatyka nr 6, 1993.

Recenzent: Dr inż. Ryszard Winiarczyk

Wpłynęło do Redakcji 17 listopada 1994 r.

Abstract

According to IT specialists, 90's will be decade of distributed systems. The aim of the work discribed in this paper is to develope subsystem which allows to treat an ordinary LAN as a distributed system. As a result users can freely distribute their tasks in all nodes and use all resources of the LAN. Remote execution of tasks is done automatically without notification of a user. Proposed subsystem is implemented in QNX operating system, which has extensions allowing distributed processing.

Subsystem of distributed processing is build on the basis of client-server model. There are two main modules in this subsystem. The first one called "dsh" (distributed shell) is a special shell which "combines" all nodes of the network and gives any user feeling of working with one, large, "single" computer (Fig.1). Every incoming task can be executed locally or in a remote node. The decision of sending task to remote execution is taken automatically on the basis of information about the state of the system. Information about workload of all active nodes is collecting in one place (Fig.2). This work is done by the second module of the subsystem called "lb" (load balancer). Module "lb" consists of several processes running in the network (Fig.4). The main process "lb" creates other processes and agregates information about state of the system. Information about workload of particular nodes is collecting by processes "lbr". Process "lbr" run in cyclic fashion a special testing process "lbr_test". Execution time of the process "lbr_test", which is "compute_bound" procedure, is the workload index of a node. This index regards computing power of a node and its load of tasks: the lower power of a node the longer time of execution and the more tasks in a node the longer time of execution of "lbr_test". Comparison of these inedexes let "lb" choose the "best" node of the network at the moment. The "best" means that this node guaranties shortest response time of particular task. Information about the number of the best node is sent to module "dsh" (Fig.3). There is another important process in module "lb": process "aktyw_wezlow". This process controls activity of all nodes. Changes in configuration of distributed system are reported to process "lb".

The subsystem of distributed processing was tested by an execution of a special process. As it is shown in Tab.1, there are benefits of using such system. Using "dsh" gives spectacular benefits in execution times of processes. This test proved that distributed processing in LAN's is possible and gives a lot of profits: shortest execution time of tasks, balance of the workload in the system and better use all resources. It is important to remind that all this things are done automatically, without notifying a user.