ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: INFORMATYKA z. 35 Nr kol. 1402

Krzysztof PIERZCHAŁA
Politechnika Śląska, Instytut Informatyki

ANALIZA PORÓWNAWCZA WYKORZYSTYWANIA RÓŻNEGO TYPU STACJI ROBOCZYCH W PRZETWARZANIU ROZPROSZONYM[1]

Streszczenie. W artykule zaprezentowano kryteria oraz sposób porównania różnych stacji roboczych w przetwarzaniu rozproszonym. Przeanalizowano straty mocy obliczeniowej spowodowane przełączaniem kilku zadań pracujących na jednym komputerze, czas powoływania nowego zadania, opóźnienia wnoszone do transmisji oraz przepustowość łącza danych dla różnych metod przesyłu. Przedstawiono wyniki doświadczeń przeprowadzonych w heterogenicznej sieci komputerowej.

COMPARATIONAL ANALYSE OF USAGE DIFFERENT WORKSTATIONS IN DISTRIBUTED COMPUTING

Summary. Criteria and metod of comparision different workstations in distributed computing is presented in the paper. Computational power lose connected with switching between tasks working on the same computer, time of creation new task, transmision latency and data link througput for different sand types are analyzed. Results of experiments taking place in heterogenical computer network are presented.

1. Wstęp

Ciągły wzrost liczby komputerów, w tym również komputerów połączonych za pomocą sieci komputerowych, oraz zwiększające się zapotrzebowanie na moc obliczeniową i wysokie ceny superkomputerów powodują powstawanie, rozwój i coraz szersze wykorzystanie systemów pozwalających na rozpraszanie obliczeń w sieci stacji roboczych. Istniejące

[1]Praca powstała w ramach grantu KBN nr 8T11C 021 11.

i rozwijane systemy rozpraszania obliczeń w sieciach bazują na stacjach roboczych pracujących pod kontrolą systemu operacyjnego Unix.

Pierwotnie w obliczeniach rozproszonych wykorzystywane były stacje robocze. Podstawową ich wadą jest dość wysoka cena. Obecnie oprócz stacji roboczych w sieciach pracuje duża liczba komputerów klasy IBM PC. Komputery te dysponują coraz większą mocą obliczeniową, niekiedy dorównującą stacjom roboczym. Nasuwa się pytanie, czy komputery osobiste mogą być równoprawnymi węzłami w przetwarzaniu rozproszonym stanowiąc tanią alternatywę dla stacji roboczych? Potencjalnie tak - coraz więcej systemów pozwalających na przetwarzanie rozproszone posiada implementacje na komputery klasy IBM PC. Głównie są to rozwiązania bazujące na różnych wersjach systemu Unix na te komputery. Oprócz wersji na komercyjne systemy unix'owe istnieją też wersje dla najpopularniejszych niekomercyjnych wersji systemu Unix: Linux i FreeBSD. Trwają również prace nad stworzeniem wersji dla popularnych środowisk pracy, takich jak MS Windows 95 i MS Windows NT.

Celem pracy jest porównanie różnych stacji roboczych (w tym komputerów osobistych) pod kątem ich zastosowania w obliczeniach rozproszonych.

2. Kryteria wydajności stacji roboczych w przetwarzaniu rozproszonym

Naturalnym kryterium w ocenie przydatności stacji do obliczeń rozproszonych jest ogólna wydajność obliczeniowa komputera. Na tą wydajność wpływają między innymi takie parametry, jak: wydajność procesora (architektura wewnętrzna, częstotliwość zegara taktującego procesor, obecność koprocesora numerycznego), rozmiar i rodzaj pamięci operacyjnej występującej w komputerze (czas dostępu do pamięci, przeplot, występowanie i wielopoziomowość pamięci typu cache), przepustowość magistrali systemowej oraz wydajność kanałów dyskowych. Wpływ tych parametrów na ogólną wydajność obliczeń (w szczególności obliczeń rozproszonych) jest oczywisty, a wiekszość parametrów jest łatwo dostępna (podają je producenci sprzętu).

W przypadku obliczeń rozproszonych głownymi kryteriami (oprócz ogólnej wydajności systemu) są przepustowość łacza danych między zadaniami obliczeniowymi (znajdującymi się potencjalnie na różnych komputerach), opóźnienie wnoszone przez komunikację (całkowity czas potrzebny na przesłanie komunikatu), czas powoływania nowego zadania obliczeniowego w systemie oraz zmiana wydajności obliczeniowej stacji pod wpływem zwiększania liczby zadań wykonywanych przez nią równolegle (współbieżnie). Przy wykorzystywaniu do obliczeń rozproszonych komputerów połączonych siecią, z których mogą też korzystać inni użytkownicy, dodatkowym kryterium powinna być możliwość wykorzystania tego komputera

do innych prac. W przypadku kryterium przepustowości łącza danych należy zwrócić uwagę na wpływ wielkości paczki danych na przepustowość oraz na wpływ rozmieszczenia zadań na szybkość transmisji (zadania są na tym samym komputerze czy na różnych komputerach).

Można polemizować z pierwszym kryterium argumentując, że przepustowość łacza danych warunkowana jest przez przepustowość sieci komputerowej. Należy jednak zwrócić uwagę na fakt, że jest to maksymalna teoretyczna przepustowość - nie oznacza to, że stacja robocza pracując w określonym środowisku pozwalającym na rozpraszanie obliczeń jest w stanie przesyłać dane z tą szybkością. Poza tym to ograniczenie nie istnieje w przypadku, gdy oba zadania komunikujące się znajdują się na jednym komputerze.

3. Wybór testowanego środowiska

Obecnie dominującymi na świecie systemami pozwalającymi na rozpraszanie obliczeń w sieciach komputerowych są systemy PVM i MPI. Ich wysoką popularność potwierdzić mogą indywidualne grupy dyskusyjne **comp.parallel.pvm** i **comp.parallel.mpi** oraz to że stały się standardem "de facto" w przetwarzaniu rozproszonym.

Jako środowisko badań został wybrany system PVM. Posiada on implementację dla większości popularnych komputerów (od IBM PC do CRAY'a). W przeciwieństwie do swojego głownego konkurenta, systemu MPI, posiada on tylko jedną rozwijaną implementację, podczas gdy system MPI posiada trzy (różniące się) implementacje typu public-domain i kilka implementacji komercyjnych.

Jako testowe stacje robocze zdecydowano się wykorzystać stacje robocze firm Sun i HewletPackard oraz komputery klasy IBM PC pracujące pod kontrolą popularnych darmowych systemów operacyjnych Linux i FreeBSD. Na wszystkich komputerach (z wyjątkiem HP) wykorzystywany był kompilator GNU C 2.7.2.1. Na wszyskich komputerach działał PVM w wersji 3.3.10.

Komputery biorące udział w testach miały następujące parametry:

- SparcClassic - procesor microSparc 50MHz, 32 MB pamięci, SunOS 4.1.3
- SparcStation 10 - procesor superSparc 30 MHz, 32 MB pamięci, SunOS 4.1.3
- HP 750/99 (9000/755) - procesor HP-PA 99MHz, 128 MB pamięci, HPUX 9.05
- Linux - procesor Intel Pentium 75 MHz ,16MB pamięci, Linux (RedHat 4.2)
- FreeBSD - procesor Intel Pentium 90 MHz, 16 MB pamięci, FreeBSD 2.2.2

Planowane było również przetestowanie popularnych komercyjnych środowisk pracy MS Windows NT i MS Windows 95. W trakcie prac okazało się, że rozwijane właśnie

implementacje systemu PVM dla tych środowisk są jeszcze niedopracowane. W ramach badań przetestowano dwie implementacje systemu PVM dla tych środowisk:

- WPVM 2.0 (popularnie zwanej "portugalską") w wersjach dla kompilatora MS Visual C++ oraz dla Borland C++ - system WPVM nie uruchamiał się, a środowisko pracy informowało o próbach zapisu w niedozwolony obszar pamięci,
- WinPVM (oficjalna wersja twórców systemu PVM - Oak Ridge National Laboratory) w wersjach od beta2 do beta4 - system nie umożliwiał połączenia do i z innych komputerów oraz nie udało się uruchomić dostarczonych z nim programów przykładowych. Prawdopodobną przyczyną były kłopoty z warstwą sieciową pomimo zainstalowania zalecanego przez twórców systemu pakietu firmy Ataman umożliwiającego wykorzystywanie tzw. r-rozszerzeń (rsh, rexec, rlogin i RCP) w środowiskach 32-bitowych firmy Microsoft.

Podjęta została również próba zaimplementowania systemu PVM dla tych środowisk korzystając z pakietu narzędzi GNU, udostępniających środowisko programowe zbliżone do środowiska systemu Unix (narzędzia systemowe, kompilator języka C z zestawem plików nagłówkowych - środowiska firmy Microsoft nie udostępniają interfejsu do funkcji systemowych). Okazało się, że pomimo bazowania na tej samej, co w systemach Unix, koncepcji gniazdek (sockets) - różnice implementacyjne są na tyle duże, że nie udało się doprowadzić do działania tej implementacji.

Twórcy systemu PVM zapewniają, że środowiska firmy Microsoft będą zaimplementowane w nowej wersji systemu PVM, która ma być niedługo dostępna w Internecie (w najbliższym czasie ma być udostępniona wersja finalna nowego systemu PVM 3.4 posiadająca wsparcie dla środowisk firmy Microsoft).

4. Testowanie typowych operacji występujących w przetwarzaniu rozproszonym

W ramach badań zostały napisane i uruchomione programy pozwalające na porównanie stacji roboczych według wcześniej ustalonych kryteriów. Powstały programy mierzące czas przesyłu pustego komunikatu między zadaniami, czas powołania pojedynczego zadania, czas przesłania paczki (paczek) danych o określonych rozmiarach między zadaniami oraz program pozwalający oszacować narzuty systemu operacyjnego przy równoległym uruchamianiu kilku zadań na jednym komputerze.

Planowano też napisanie programów pomocniczych służących do symulowania obciążenia stacji roboczej oraz sieci komputerowej. W trakcie testów stwierdzono, że same programy testowe powodują duże obciążenie stacji roboczych. Poza tym nie udało się stworzyć

wyizolowanego fragmentu sieci, co powodowało zmienne w czasie obciążenia stacji roboczych i sieci jako takiej, co nie umożliwiało zmiany tego obciążenia w sposób kontrolowany. W celu otrzymania wiarygodnych wyników zdecydowano się na wykonywanie testów w czasie najmniejszego obciążenia sieci oraz na wielokrotne powtarzanie testów w celu zredukowania krótkotrwałych zmian obciążenia.

Decydując się na pomiary określonych czasów wybrano takie, które pozwalają po niewielkich przeliczeniach uzyskać informacje odpowiadające kryteriom wskazanym w poprzednim rozdziale.

Przepustowość łącza między zadaniami można obliczyć jako iloraz rozmiaru paczki lub iloczynu rozmiaru paczki i liczby paczek (należy uwzględnić, że rozmiar paczki jest to liczba danych typu integer - co dla badanych systemów oznacza konieczność przemnożenia rozmiaru paczki przez 4 (aby uzyskać przepustowość w bajtach/sec) lub 32 (dla bitów/sec)) i czasu przesyłu tej paczki (wyrażonego w sekundach). Przy czym można wyróżnić dwa rodzaje przepustowości: przepustowość, jaką widzi zadanie wysyłające informację i rzeczywistą przepustowość łącza. Pierwsza przepustowość jest nie mniejsza od drugiej - wynika to z faktu, że funkcja wysyłająca informacje może zwrócić kontrolę (i w przypadku systemu PVM zwraca) do programu przed całkowitym wysłaniem danych do drugiego zadania. Rzeczywistą przepustowość łącza możemy uzyskać mierząc czas od początku wysłania danych do uzyskania potwierdzenia (pustym komunikatem) z drugiego zadania informującego, że drugie zadanie dostało już wszystkie dane. Ponieważ system PVM umożliwia dwa typy przygotowania danych do wysłania (z przeniesieniem danych do wewnętrznego bufora systemu PVM "PvmDataDefault" i z bez przeniesienia danych do tego bufora - dane zostają wysłane bezpośrednio z miejsca, w którym się znajdują "PvmDataInPlace"), zostało to uwzględnione w programie testowym. Należy tu zauważyć różnicę między buforami systemu PVM a buforami komunikacyjnymi wykorzystywanymi przez system operacyjny.

Opóźnienie w komunikacji między zadaniami jest to bezpośrednio czas przesyłu pustego komunikatu między zadaniami.

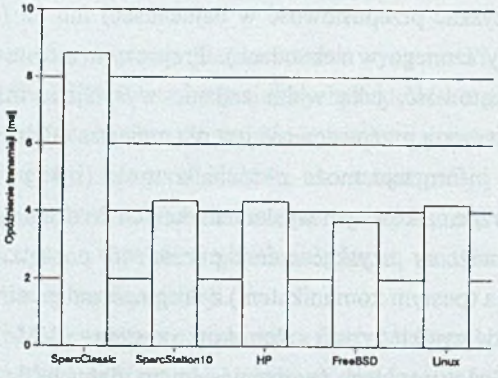
Czas powołania pojedynczego zadania był uzyskiwany bezpośrednio w następujący sposób: powoływane zostało zadanie, które zaraz po starcie przesyłało pusty komunikat do zadania nadzorującego. Mierzony był czas od wywołania funkcji do uzyskania od niej komunikatu - czas powołania pojedynczego zadania jest równy zmierzonemu czasowi pomniejszonemu o czas przesyłu pustego komunikatu.

Ocena zmiany wydajności stacji roboczej pod wpływem zwiększania liczby zadań wykonywanych na niej równolegle została zmierzona poprzez pomiar czasu wykonania zadania o określonej złożoności (konkretnie wyliczenia fraktala Mandelbrota) w jednym kawałku oraz w kilku częściach wykonywanych równolegle. W przypadku idealnego systemu operacyjnego czasy te powinny być sobie równe, w przypadku rzeczywistego systemu

operacyjnego czasy te będą sie różnić między sobą pozwalając ocenić procentowe narzuty na zwiększanie liczby zadań działających w systemie.

5. Wyniki

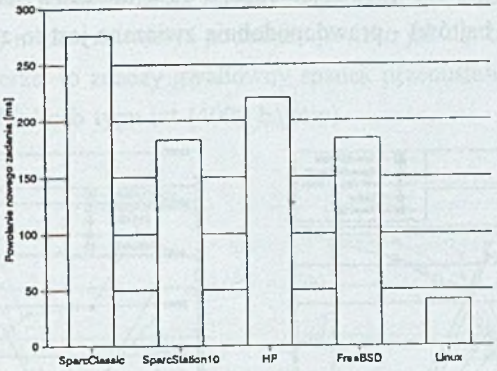
Opóźnienie transmisji danych przesyłanych między różnymi komputerami (rys. 1) jest porównywalne dla wszystkich testowanych komputerów. Gorszy wynik komputera SparcClassic można tłumaczyć największym obciążeniem - znajdował się na nim pracujący serwer bazy danych.



Rys. 1. Opóźnienia transmisji danych między zadaniami na różnych komputerach
Fig. 1. Delay of data transmision between tasks located on different computers

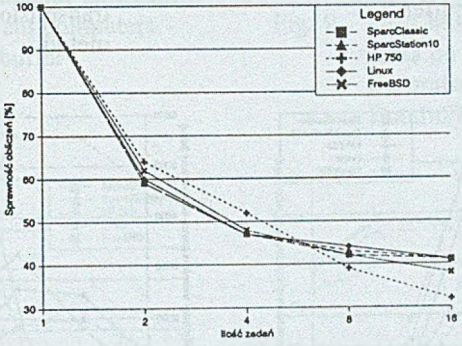
Czas powołania nowego zadania (rys. 2) jest porównywalny we wszystkich systemach. Wyjątkiem jest tu system Linux, w którym czas powołania nowego zadania jest średnio pięć razy krótszy. Oznacza to, że w systemach rozproszonych, w których problem jest rozbity na małe podzadania - co oznacza dużą ilość małych programów (zagadnienia drobnoziarniste) - czas rozwiązania tego problemu w sieci komputerów z systemem Linux powinien być krótszy niż w przypadku innych rozwiązań.

Analizując zmianę sprawności obliczeń w funkcji ilości równolegle powołanych zadań (rys. 3), można zaobserwować gwałtowny spadek sprawności przy małej ilości zadań. Przy większej ilości zadań systemy stabilizują sprawność obliczeń na poziomie około 40% wartości początkowej. Wyjątkiem jest tu komputer HP, w którego przypadku widać trudności z równoczesnym uruchamianiem kilku zadań.



Rys. 2. Czas powołania nowego zadania
Fig. 2. Time of creation new task

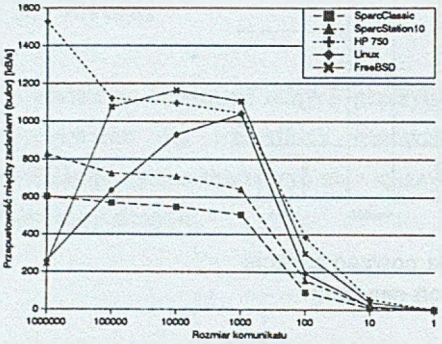
Z analizy zmiany sprawności wypływa wniosek, że dla systemu PVM należy tak pisać programy, aby powoływać minimalną ilość podzadań (najlepiej tylko jedno) na każdym komputerze wchodzącym w skład maszyny wirtualnej.



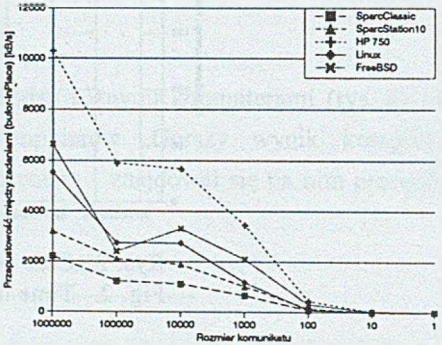
Rys. 3. Sprawność obliczeń w zależności od ilości zadań wykonywanych równolegle
Fig. 3. Computational efficiency in function of parallel running tasks

Przy analizie przepustowości łącza danych między zadaniami znajdującymi się na jednym komputerze (rys. 4-7) można zaobserwować bardzo różne zachowania poszczególnych komputerów. Wszystkie testowane komputery wykazują gwałtowny spadek przepustowości dla komunikatów o rozmiarze pomiędzy 1000 a 100 liczb typu int (4000 - 400 bajtów). Związane jest to z maksymalnym rozmiarem pojedynczego pakietu w sieci. Zauważyć też można odmienne zachowanie komputerów klasy IBM PC, które w tym przypadku wykazują

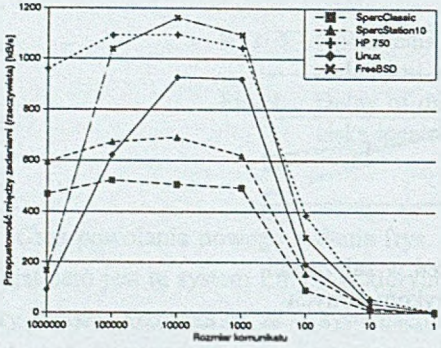
się maksymalną przepustowością dla komunikatów o rozmiarach z zakresu 100000-1000 liczb typu int (400000-4000 bajtów) - prawdopodobnie związane jest to ze sposobem gospodarki pamięcią.



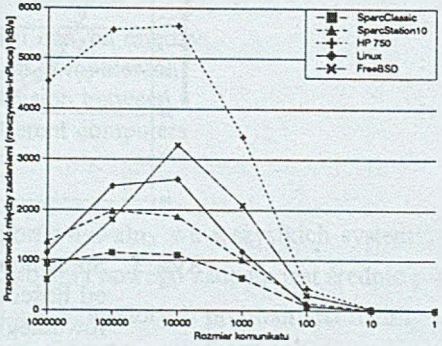
Rys. 4. Przepustowość łącza między zadaniami znajdującymi się na jednym komputerze - przenoszenie danych do bufora
Fig. 4. Data link throughput between tasks on the same computer - transmision to buffer



Rys. 5. Przepustowość łacza między zadaniami na jednym komputerze - przenoszenie danych do bufora, metoda DataInPlace
Fig. 5. Data link throughput between tasks on the same computer - transmision to buffer, DataInPlace metod

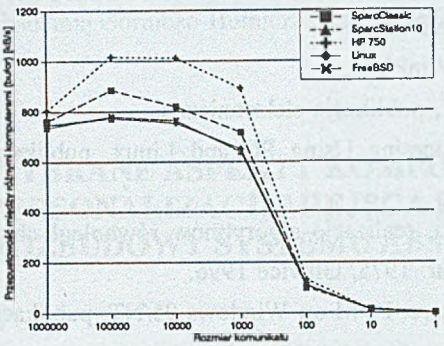


Rys. 6. Przepustowość łacza między zadaniami znajdującymi się na jednym komputerze
Fig. 6. Data link throughput between tasks on the same computer - true transmision



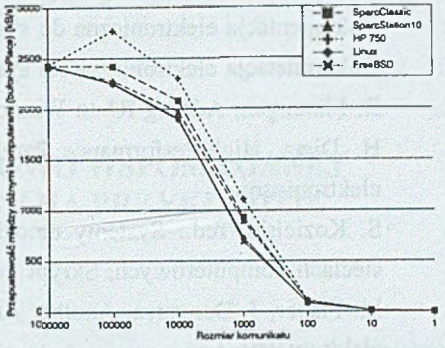
Rys. 7. Przepustowość łacza danych między zadaniami znajdującymi się na jednym komputerze - metoda DataInPlace
Fig. 7. Data link throughput between tasks on the same computer - true transmision, DataInPlace method

W przypadku analizy łącza danych między zadaniami znajdującymi się na różnych komputerach (rys. 8-11) można zauważyć podobne zjawisko jak dla łącza między zadaniami na tym samym komputerze, to znaczy gwałtowny spadek przepustowości dla komunikatów o rozmiarze poniżej 1000 liczb typu int (4000 bajtów).



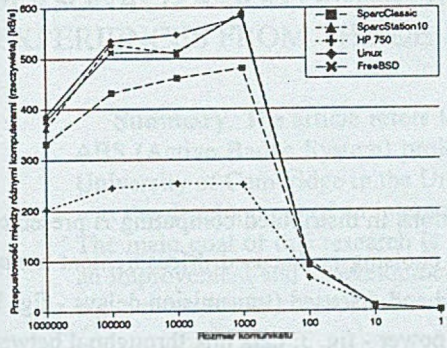
Rys. 8. Przepustowość łącza danych między zadaniami znajdującymi się na różnych komputerach - przenoszenie danych do bufora

Fig. 8. Data link throughput between tasks on different computers - tranmision to buffer



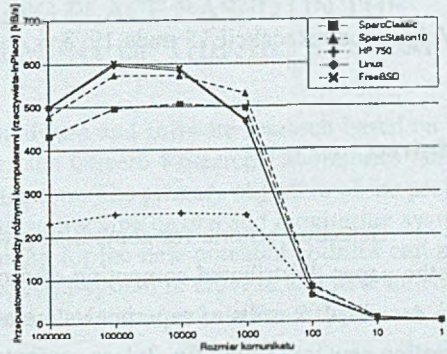
Rys. 9. Przepustowość łącza między zadaniami znajdującymi się na różnych komputerach - przenoszenie danych do bufora, metoda DataInPlace

Fig. 9. Data link throughput between tasks on different computers - transmission to buffer, DataInPlace method



Rys. 10. Przepustowość łącza między zadaniami znajdującymi się na różnych komputerach

Fig. 10. Data link throughput between tasks on different computers - true transmision



Rys. 11. Przepustowość łącza między zadaniami znajdującymi się na różnych komputerach - metoda DataInPlace

Fig. 11. Data link througput between tasks on different computers - true transmision, DataInPlace method

LITERATURA

1. Dokumentacja elektroniczna do systemu PVM.
2. Praca zbiorowa, PVM: Parallel Virtual Machine A Users' Guide and Tutorial for Networked Parallel Computing, The MIT Press 1994.
3. Dokumentacja elektroniczna do systemu WPVM.
4. Dokumnetacja elektroniczna do systemu WinPVM.
5. R. Flannigan, Adding R* to Windows NT, publikacja elektroniczna.
6. H. Dietz, High-Performance Parallel Procesing Using PC and Linux, publikacja elektroniczna.
7. S. Kozielski, red.: Systemy umożliwiające realizację algorytmów równoległych w sieciach komputerowych, Skrypt Pol. Śl. nr. 1975, Gliwice 1996.
8. M. Fisher, J. Dongarra, Another Architecture: PVM on Windows 95/NT, publikacja elektroniczna.
9. Dokumentacja elektroniczna do pakietu CYGWIN.
10. Dokumentacja elektroniczna do pakietu NetPerf.
11. Dokumentacja elektroniczna do pakietu ParkBench.
12. Dokumentacja elektroniczna do pakietu pvmBench.
13. The Ataman TCP Remote Logon Services User's Manual, publikacja elektroniczna.

Recenzent: Dr inż. Maciej Bargielski

Wpłynęło do Redakcji 15 maja 1998 r.

Abstract

Selection criteria for comparising workstations in distributed computing is presented in the paper. Distributed computing environment choising is discused. Measurement metod are described. Results of experiments are presented and discused (transmision delays - fig. 1; creation new task time - fig. 2; lose computational power - fig. 3; data link throughput betwen tasks on the same computer - fig. 4-7; data link throughput between tasks on different computers fig. 8-11).